

“YoungProgramming–Una estrategia, colaborativa y lúdica, en apoyo al Aprendizaje de la Programación en Equipos conformados por Jóvenes”



*Monografía para optar al título de
Ingeniero de Sistemas*

**Karen Johanna Reyes Rivera
Aldemar Saavedra Buitrago**

Director: Ph.D. César Alberto Collazos Ordoñez
Codirector: Ph.D. Julio Ariel Hurtado Alegría.

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I + D en Ingeniería de Software – IDIS
Línea de Investigación en Ingeniería del Software
Popayán, Septiembre de 2016**

AGRADECIMIENTOS

A Dios, por darme la vida y nunca dejarme sola. A mi madre, Nelly Rivera por su amor de siempre y su apoyo incondicional en todas las decisiones de mi vida.

A mis hermanas Paola y Adriana, a mi hermano Santiago, a mis abuelos y a mi tío Carlos A. Rivera quienes siempre me han acompañado a lo largo de mi vida.

A mi compañero de tesis, Aldemar Saavedra, por su comprensión, dedicación y compañía.

A la Universidad del Cauca y docentes por las enseñanzas y momentos vividos, determinantes en mi vida profesional.

Karen Johanna Reyes Rivera

Hoy que este logro se hace realidad, agradezco a Dios, por sus bendiciones y por aquellas personas que con su influencia directa o indirectamente estuvieron presentes en las distintas etapas de mi carrera.

A mi Madre Maritza Buitrago, por su amor incondicional, apoyo y gran trabajo como madre.

A mi Padre Aldemar Saavedra Ospina, de quien aprendí la integridad y tenacidad, cualidades presentes en todas sus acciones diarias.

A mi hermana Valentina por su compañía y apoyo a lo largo de este camino.

Aldemar Saavedra Buitrago

Agradecimientos Especiales

Gracias al Ingeniero César Collazos y Julio Ariel Hurtado por la dirección de este proyecto, por la disposición que tuvieron y la ayuda en los momentos adecuados.

Al Ingeniero René Zúñiga y demás docentes de la Institución Educativa Técnico Industrial Sede Principal por su apoyo en la investigación.

Al Ingeniero Freddy Muñoz por su apoyo en el trabajo de investigación.

Y muchas gracias a todas aquellas personas que colaboraron o participaron, de una u otra forma, en la realización de este trabajo.

Tabla de contenido

ÍNDICE DE TABLAS.....	V
ÍNDICE DE FIGURAS.....	VI
Capítulo 1.....	1
1.1 INTRODUCCIÓN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	2
1.3 PROPUESTA.....	3
1.4 OBJETIVOS.....	4
1.4.1 Objetivo general.....	4
1.4.2 Objetivos específicos.....	4
1.5 METODOLOGÍA.....	4
1.5.1 Fase de Exploración.....	4
1.5.2 Fase de Formulación - Planificación.....	5
1.5.3 Fase de Ejecución.....	5
1.5.4 Fase de Consolidación.....	5
1.5.5 Fase de Divulgación y Documentación.....	5
1.6 APORTES.....	6
1.6.1 En el ámbito académico.....	6
1.6.2 Investigación.....	6
1.6.3 Desarrollo.....	6
1.7 ORGANIZACIÓN DEL DOCUMENTO.....	6
2 Capítulo 2.....	8
2.1 REVISIÓN DE LA LITERATURA Y TRABAJOS RELACIONADOS.....	8
2.1 Aprendizaje Colaborativo (AC).....	8
2.2 Distinción entre aprendizaje cooperativo y colaborativo.....	9
2.3 Pensamiento Computacional.....	10
2.4 CSCL dentro de la educación.....	10
2.5 Enseñanza de la Programación.....	11
2.6 Lúdica Como Estrategia Didáctica.....	11
2.7 Efectos del Juego en la Enseñanza.....	11
2.7.1 Niveles de aplicación de los juegos.....	13

2.7.2	Elementos de los Juegos	15
2.8	Actividad Colaborativa.....	17
2.8.1	Características de una actividad colaborativa	17
2.9	Diseño de Procesos Colaborativos	20
2.9.1	Actividad.....	20
2.9.2	Roles	20
2.9.3	Recursos o Materiales	20
2.9.4	Grupos	20
2.9.5	Objetivos	20
2.9.6	Reglas de juego.....	20
2.9.7	Criterios de éxito	21
2.10	Modelado de Procesos Colaborativos	21
2.10.1	CIAM una propuesta para modelar procesos colaborativos.....	21
2.10.2	AMENITIES una metodología para el análisis y diseños de sistemas cooperativos ..	22
2.11	Mecanismos para fomentar el Aprendizaje Colaborativo.....	24
2.12	CHILDPROGRAMMING	25
2.13	SCRATCH	27
2.14	Trabajos Relacionados.....	28
3	Capítulo 3	33
3.1	MÉTODO DE APLICACIÓN DE LA LÚDICA Y LA COLABORACIÓN	34
3.2	LudiCoP (Ludic-Collaborative Programming). Aspectos Colaborativos	39
3.3	LudiCoP (Ludic-Collaborative Programming). Aspectos Lúdicos.	42
3.4	Modelamiento de LudiCoP a través de CIAM.....	43
4	Capítulo 4	54
4.1	CONTEXTO DEL MODELO PROPUESTO.....	55
4.1.1	Definición del Proceso YoungProgramming.....	55
4.1.2	El rol de LudiCoP en YoungProgramming.....	56
4.1.3	Las Mecánicas de Juego en YoungProgramming.....	56
4.2	ARQUITECTURA CONCEPTUAL DE YOUNGPROGRAMMING	56
4.2.1	Actores	57
4.2.2	Roles	57
4.2.3	Mecánicas de Juego Dimensión Lúdica	59

4.3	PROCESO YOUNGPROGRAMMING.....	61
4.3.1	Ciclo de vida del proceso YoungProgramming.....	61
4.3.2	Fase pre-proceso.....	62
4.3.3	Fase proceso.....	63
4.3.4	Fase post-proceso.....	64
5	Capítulo 5.....	66
5.1	METODOLOGÍA.....	67
5.1.1	Instrumentos de Evaluación.....	68
5.2	CONTEXTO DE LA INVESTIGACIÓN.....	69
5.3	ESTUDIOS DE CASO.....	70
5.3.1	ESTUDIO DE CASO 1: APLICANDO YOUNGPROGRAMMING Y LUDICOP EN SU VERSIÓN INICIAL (EC1).	70
5.3.2	ESTUDIO DE CASO 2: APLICANDO YOUNGPROGRAMMING Y LUDICOP EN UNA VERSION MEJORADA (EC2).	85
6	Capítulo 6.....	98
6.1	CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	98
6.2	RESUMEN DEL PROYECTO.....	99
6.3	CONCLUSIONES.....	99
6.4	RECOMENDACIONES.....	100
6.5	TRABAJO FUTURO.....	100
7.	REFERENCIAS BIBLIOGRAFICAS.....	102

ÍNDICE DE TABLAS

Tabla 1 – Diferencia entre Grupos Colaborativos y Grupos Tradicionales	9
Tabla 2 – Aspectos que Mejora el Juego	13
Tabla 3 – Comparación de modelos de la revisión de la literatura	31
Tabla 4 – Modelo YoungProgramming	32
Tabla 5 – Comparación de técnicas de Aprendizaje Colaborativo	37
Tabla 6 – Procedimiento de técnica de AC “LudiCoP”	42
Tabla 7 – Dinámica “Reglas” dentro de LudiCoP	42
Tabla 8 – Dinámica “Competencia” dentro de LudiCoP	43
Tabla 9 – Componente “Recompensa” dentro de LudiCoP	43
Tabla 10 –Tabla de Partición CIAM elaborada a partir de Tabla 5	45
Tabla 11 – Modelo de Responsabilidades del <i>Rol Profesor</i>	46
Tabla 12 – Modelo de Responsabilidades del <i>Rol Estudiante</i>	47
Tabla 13 – Modelo de Responsabilidades del <i>Rol Observador/Investigador</i>	48
Tabla 14 –Comparación de Roles de YoungProgramming	58
Tabla 15 – Puntos en la Dimensión Lúdica YoungProgramming	59
Tabla 16 – Desafíos en la Dimensión Lúdica YoungProgramming	59
Tabla 17 – Tablas de Clasificación en la Dimensión Lúdica YoungProgramming	60
Tabla 18 – Beneficios en la Dimensión Lúdica YoungProgramming	60
Tabla 19 – Tiempos Regresivos en la Dimensión Lúdica YoungProgramming	61
Tabla 20 – Actividades Estudio de Caso número uno (EC1) – Guía LudiCoP	74
Tabla 21 – Tabla de indicadores y mediciones	74
Tabla 22 – Productividad grupo experimental EC1	77
Tabla 23 – Productividad grupo control EC1	77
Tabla 24 – Calidad grupo experimental EC1	78
Tabla 25 – Calidad grupo control EC1	79
Tabla 26 – Comportamiento grupo experimental EC1	80
Tabla 27 – Comportamiento grupo de control EC1	81
Tabla 28 – Actividades Estudio de Caso número dos (EC2) – Guía LudiCoP	89
Tabla 29 – Tabla de indicadores y mediciones	90
Tabla 30 – Productividad grupo final EC2	91
Tabla 31 – Calidad grupo final EC2	92
Tabla 32 – Comportamiento grupo final EC2	93

ÍNDICE DE FIGURAS

Figura 1 – Modelo conceptual de la revisión de la literatura	8
Figura 2 – Etapas de la metodología CIAM	21
Figura 3 – Ciclo de vida de ChildProgramming	26
Figura 4 – Modelo inicial planteado de ChildProgramming	27
Figura 5 – Contextualización del capítulo 3	33
Figura 6 – Esquema general de aplicación de Aprendizaje Colaborativo	34
Figura 7 – Sociograma de LudiCoP	44
Figura 8 – Modelo de Inter-Acción de LudiCoP	49
Figura 9 – Tarea cooperativa 2.1	50
Figura 10 – Tarea colaborativa 2.2	50
Figura 11 – Tarea cooperativa 2.3	51
Figura 12 – Tarea cooperativa 2.4	51
Figura 13 – Tarea colaborativa 2.5	52
Figura 14 – Tarea colaborativa 2.7	52
Figura 15 – Tarea colaborativa 3.1	53
Figura 16 – Tarea cooperativa 3.2	53
Figura 17 – Tarea cooperativa 3.3	53
Figura 18 – Arquitectura de ChildProgramming	55
Figura 19 – Arquitectura de YoungProgramming	57
Figura 20 – Ciclo de Vida del Proceso YoungProgramming	61
Figura 21 – Procedimiento metodológico de los estudios de caso	68
Figura 22 – Instrumentos de Evaluación	69
Figura 23 – Productividad del grupo experimental EC1	77
Figura 24 – Productividad del grupo de control EC1	78
Figura 25 – Calidad del grupo experimental EC1	79
Figura 26 – Calidad del grupo de control EC1	80
Figura 27 – Comportamiento del grupo experimental EC1	81
Figura 28 – Comportamiento del grupo de control EC1	82
Figura 29 – Porcentaje por cursos (Experimental y Control)	82
Figura 30 – Porcentaje de evaluación de aprendizaje por cursos	83
Figura 31 – Porcentaje de evaluación de aprendizaje de mujeres por cursos	84
Figura 32 – Resultados encuestas (Experimental EC1)	84
Figura 33 – Productividad grupo final EC2	92
Figura 34 – Calidad grupo final EC2	93
Figura 35 – Comportamiento grupo final EC2	94
Figura 36 – Porcentaje del curso final EC2	94

Capítulo 1

1.1 INTRODUCCIÓN

Hoy en día los estudios sobre el mercado de trabajo muestran que la oferta de ingenieros con conocimientos en programación no está cerca de satisfacer la demanda. Las empresas, sin importar su tamaño, necesitan de personas con conocimientos en programación que les ayuden a desarrollar las aplicaciones informáticas que les permitan crecer en sus mercados [72].

Aunque en Colombia las instituciones de educación superior ofrecen 93 programas de ingeniería, el país no cuenta con todos los ingenieros que su desarrollo demanda; el déficit resulta notorio en las áreas de tecnología [72].

La encuesta anual de escasez de talentos, de ManpowerGroup, señala también que el puesto más difícil de cubrir es el de los ingenieros, que a su vez resulta ser el cargo de mayor demanda en las empresas colombianas, principalmente relacionado con tecnología y sistemas de información e innovación. Los estudios estiman que hay un déficit de 15.000 ingenieros de sistemas y la cifra será de 93.000 en 2018, si no se toman medidas; además se espera también que en el año 2018 las firmas desarrolladoras de software se dupliquen y que los 39.000 empleos relacionados con las TIC se multipliquen por tres [72].

La tecnología cada vez es más accesible a las personas, después de años de trabajo con computadores y tecnologías, finalmente se ha llegado al punto donde éstas han permeado casi todos los aspectos de la vida diaria de los seres humanos. Para poder adaptarse a este mundo cambiante las empresas de software y los ingenieros de desarrollo de software deben estar en capacidad de adquirir continuamente nuevos conocimientos y capacidades que estén a la altura de cubrir esta demanda. Una estrategia es involucrar desde el colegio y primeros semestres de educación superior a los jóvenes ya que ellos son la generación que se encargará de suplir estos problemas de escasez de personas capacitadas.

Las dificultades en la enseñanza/aprendizaje de la programación ha sido un problema recurrente en los últimos 25 años en nuestro país como en el mundo entero; la estigmatización de los jóvenes que piensan que la programación y las carreras relacionadas con esta área son difíciles, la dificultad para desarrollar el pensamiento lógico y computacional para resolver problemas y los problemas de motivación de los estudiantes son algunos de los tantos obstáculos o conflictos a los que se ven enfrentados quienes inician sus estudios universitarios de programación. Es por esto que a lo largo del tiempo se han propuesto numerosas soluciones sin que ninguna haya resultado cien por ciento efectiva y actualmente se vive en la búsqueda de estrategias metodológicas que las solucionen. A través de la experiencia con algunas poblaciones, el grupo IDIS explora las técnicas, prácticas y uso de las tecnologías que los equipos de trabajo del futuro tendrán a través del estudio del trabajo en equipos de jóvenes de hoy en día, particularmente en como desarrollan un pensamiento computacional desde los últimos años de colegio y primeros semestres de educación superior. Por medio de esta investigación se pretende proponer un proceso de desarrollo de software para el apoyo al aprendizaje de la programación en equipos conformados por jóvenes como estrategia para el aprendizaje y la enseñanza de la programación en los años en los que mayor deserción estudiantil se presenta.

1.2 PLANTEAMIENTO DEL PROBLEMA.

En la enseñanza y el aprendizaje de la programación, se puede observar que se tienen grandes dificultades y que los estudiantes no se encaminan mucho en la programación de computadoras durante sus estudios universitarios. Uno de los principales puntos de preocupación es el de cómo capacitar y hacer que aprendan a construir aplicaciones de la vida real de una mejor manera utilizando un lenguaje de programación. La razón principal del desinterés en la programación se identifica como la falta de motivación y compromiso de los estudiantes en el aprendizaje de los conceptos de programación [2].

La motivación juega un papel muy relevante en el proceso de enseñanza-aprendizaje. Aunque existen numerosos estudios sobre la importancia de la motivación, hay un “escaso conocimiento científico sobre los factores que afectan a la motivación, disfrute y satisfacción” [1]. La lúdica motiva al estudiante para la interacción de sus significados con otros y para que aplique los nuevos conocimientos en otros contextos. Esto genera en el estudiante una seguridad afectiva debido a que el estudiante integra el pensamiento, la actuación y la afectividad, dándole significado a su experiencia[17].

Hay una gran variedad de posibles motivaciones para aprender a programar: perseguir la programación como una carrera, para construir software personalizado para su uso personal, para explorar ideas en otras áreas temáticas, para aprender a resolver problemas de una manera estructurada y lógica, etc. [10]. Estos problemas se pueden resolver a través del Pensamiento Computacional. El pensamiento computacional permite reforzar el razonamiento lógico y el razonamiento analítico, donde equipados con dispositivos de cómputo, usamos nuestra inteligencia para hacer frente a problemas que no se podían solucionar antes de la era de la informática y construir sistemas con funcionalidad limitada sólo por nuestra imaginación. El Pensamiento computacional está reformulando un problema aparentemente difícil en uno solo que sabemos cómo resolver, tal vez por la reducción, la incorporación, transformación, o simulación [11].

Para enseñar y generar habilidades de programación se debe destacar la importancia del trabajo en equipo para los jóvenes [1]. Aprender colaborativamente implica la solución conjunta de problemas, permitiendo que el aprendizaje surja como una consecuencia [3]. Para la enseñanza de la programación, han surgido una gran variedad de experiencias y herramientas, como por ejemplo: Karel¹, Alice², Scratch³, Kodu⁴, el lenguaje de programación LOGO⁵ y los mecanismos diseñados por RoboTech Center⁶, que ofrece cursos de diseño de aplicaciones de modelamiento en 3D, tendientes a ejercitar la imaginación. Sin embargo, ninguno de estos modelos plantea estrategias de trabajo colaborativo y unos pocos aplican el concepto de la lúdica [1].

Los mecanismos de aprendizaje son los responsables de activar los procesos cognitivos que producen el aprendizaje en los estudiantes que forman parte de una situación de aprendizaje colaborativo [1]. Lamentablemente, estos mecanismos no siempre se activan en la colaboración [3]. Por esta razón autores como Johnson & Johnson [4] han definido la

¹Karel: Es una herramienta que promueve la creatividad y la lógica de una manera ordenada. Lo que crea una base fuerte para el aprendizaje de la programación. Las instrucciones que utiliza son sencillas, por lo que no requieren un estudio profundo para su comprensión

²Alice: Herramienta de enseñanza gratuita diseñada para ser el primer contacto con la programación orientada a objetos.

³Scratch: <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>

⁴Kodu: <http://research.microsoft.com/en-us/projects/kodu>

⁵Logo: Lenguaje de programación, propuesto por Papert en 1980 y que se acompaña de una serie de reflexiones sobre el uso de la computadora en la educación.

⁶RoboTech Center: <http://robotechcenter.com/innovate/>

existencia de algunos elementos imprescindibles que se deben dar en la colaboración para que el aprendizaje tenga lugar de manera efectiva. El número exacto, nombre y orden de estos elementos varía de uno a otro autor. Sin embargo, casi todos coinciden en que, de una manera u otra, los elementos listados a continuación son esenciales: Interdependencia Positiva, Igual Participación, Responsabilidad Individual [5].

En este sentido, el paradigma educativo del Aprendizaje Colaborativo Soportado por Computador (*CSCL, Computer Supported Collaborative Learning*), pretende aprovecharse del efecto sinérgico que supone el hecho de que varios estudiantes colaboren para resolver un determinado problema [15]. Dado que no se tiene certeza sobre qué problemas van a enfrentar en unos años los jóvenes de hoy, una de las pocas cosas efectivas que se puede hacer es enseñarles cómo expresar sus problemas en el “lenguaje del computador”, de manera que puedan resolver ellos mismos los problemas que nosotros no somos capaces de prever [9].

1.3 PROPUESTA.

Para realizar una enseñanza de la programación y facilitar la práctica del desarrollo de software en niños, el grupo IDIS ha propuesto un modelo conceptual llamado *ChildProgramming* [1], el cual plantea estrategias de trabajo colaborativo y aprovecha los enfoques ágiles para el desarrollo de software basado en paradigmas modernos, así como prácticas cognitivas [1].

El modelo *ChildProgramming* sale a partir de una evaluación de una serie de prácticas que adoptan otros modelos pedagógicos tales como el modelo constructivista de Piaget [27], el sociocultural de Lev Vygotsky [28], el modelo de aprendizaje significativo de David Ausubel y el modelo activista de María Montessori [29]. Estas prácticas finales son analizadas y caracterizadas dentro del proceso *ChildProgramming*.

Cómo parte de *ChildProgramming*, se ha definido el Proceso *ChildProgramming*, el cual guía a los equipos para alcanzar sus objetivos de desarrollo, brindando un conjunto de prácticas colaborativas, ágiles y cognitivas para el desarrollo de las actividades de construcción de software [1]. Dentro del cual se establecen cuatro (4) roles, divididos en: tres (3) roles participantes (Profesor, guía del equipo, equipo de trabajo) y un (1) rol externo (Investigador/observador), y además una serie de prácticas cognitivas, ágiles y colaborativas en cada etapa de desarrollo del proceso.

Sin embargo las actividades del *Proceso ChildProgramming* no fueron definidas para jóvenes, y tampoco consideran el componente lúdico inicialmente planteado en el modelo conceptual. Por lo tanto, carece de los mecanismos que podrían incrementar aún más la motivación, el interés, el compromiso y por ende el desempeño de los equipos durante las actividades de desarrollo de software [1]. Tampoco, se evidencian los mecanismos de aprendizaje responsables de activar los procesos cognitivos que producen el aprendizaje en los estudiantes que forman parte de una situación de aprendizaje colaborativo. En el marco de esta iniciativa es que surge la pregunta de investigación de este proyecto: ¿Cómo lograr que los adolescentes aprendan a programar en un entorno basado en la lúdica y el aprendizaje colaborativo?

Éste proyecto busca resolver ésta pregunta mediante la adaptación del modelo *ChildProgramming* a través de la investigación y reformulación de sus prácticas, roles y artefactos, considerando aspectos claves de lúdica y colaboración que faciliten el aprendizaje de la construcción de software en equipos de desarrollo jóvenes y la búsqueda de un enfoque pedagógico más adecuado, realista y ameno, donde se promuevan sencillas pero efectivas prácticas que permitan al estudiante alcanzar un nivel de experiencia

memorable y duradera, logrando así un nivel de madurez mayor en relación con la implementación de aplicaciones.

1.4 OBJETIVOS

A continuación, se describen el objetivo general y los objetivos específicos.

1.4.1 Objetivo general

- Adaptar el proceso ChildProgramming a equipos de desarrollo pequeños conformado por jóvenes entre los 14 y 19 años a través de la reformulación de sus prácticas, roles y artefactos, y considerando aspectos claves de lúdica y colaboración acorde a este rango de edad.

1.4.2 Objetivos específicos.

- Establecer las estrategias colaborativas y lúdicas que vienen apoyando a los procesos de enseñanza-aprendizaje de la programación en jóvenes, a través de una revisión de la literatura.
- Adaptar el proceso ChildProgramming a equipos conformados por jóvenes a través de la incorporación de aspectos lúdicos y estrategias de colaboración siguiendo un enfoque empírico incremental.
- Reformular las prácticas, roles y artefactos del proceso ChildProgramming orientándolas hacia equipos conformados por jóvenes.
- Evaluar la propuesta incrementalmente a través de dos (2) estudios de caso con jóvenes de instituciones educativas del departamento del Cauca.

1.5 METODOLOGÍA

Para el desarrollo del presente trabajo de grado se seguirá la metodología de estudio de caso [18] adaptada a las necesidades de éste proyecto de investigación, Scrum como marco de gestión del trabajo de investigación [19] basada en el método original [20]. A continuación se describe la metodología como se desarrollara el proyecto:

1.5.1 Fase de Exploración

En esta fase se realizará un estudio inicial, en donde se estructurará de manera detallada el proyecto a desarrollar, aportando así, al planteamiento de estrategias y elementos necesarios para lograr los objetivos de este proyecto.

Con el fin de fortalecer la base conceptual, se hará una exploración más elaborada acerca de los antecedentes que hacen parte de los siguientes núcleos temáticos: enseñanza y aprendizaje de la programación, pensamiento computacional, lúdica y trabajo colaborativo, a través de la consulta de libros, artículos, monografías, consultas a expertos en los temas y demás fuentes de información que nos faciliten las herramientas necesarias para consolidar la investigación y así poder disponer de criterios suficientes para establecer un estudio robusto. Se incluye una planificación general inicial.

En esta fase se hace entrega del Anteproyecto de trabajo de grado y se complementa el estado del arte realizando un estudio profundo de los núcleos temáticos.

1.5.2 Fase de Formulación - Planificación

Esta fase incluye dos subfases: planeación y diseño de alto nivel/arquitectura.

➤ **Planeación:** en esta fase se crea una lista de requisitos del modelo YoungProgramming en base a los ya establecidos en el modelo existente ChildProgramming. También se incluye dentro de la planeación la definición del equipo del proyecto incluyendo la institución educativa donde se trabajará y el grupo de jóvenes con el que se contará, las herramientas y/o entornos de desarrollo software con los que se experimentará y otros recursos (equipos, materiales, etc.), entrenamiento necesario y verificación para la aceptación. Se especifica que en cada sprint⁷, se hace una actualización del nuevo modelo para ser revisada por el equipo y así lograr el objetivo en la iteración siguiente.

➤ **Estrategia:** el diseño incluye la estrategia del modelo (Modelo inicial (Mo), Estudio de caso y Modelo Incrementado (Mi)) basado en los requisitos.

1.5.3 Fase de Ejecución

Extracción, formulación, evaluación incremental del modelo “YoungProgramming”. En esta fase se dará seguimiento al cronograma, a los requisitos, a los recursos, las tecnologías y las herramientas de la puesta en práctica, se observarán y controlarán con prácticas de la metodología Scrum como lo son: pila de temas (requisitos) del modelo, reunión de planificación, ejecución del sprint, entrega y reunión de revisión del sprint. Cada Sprint incluye las siguientes actividades: la observación (extracción), la formulación del modelo (prácticas, roles y artefactos), la evaluación de la adaptación propuesta a través de estudios de caso, la integración y/o ajuste de los elementos en el ambiente de desarrollo y la retrospectiva (reflexión del trabajo de investigación).

Durante la fase de ejecución se realizarán dos (2) Sprints, donde cada uno permitirá una realimentación y refinamiento del modelo. En esta fase se entrega el modelo obtenido, y un artículo como realimentación de la experiencia el cual será enviado a una revista indexada y/o evento nacional o internacional.

1.5.4 Fase de Consolidación

Esta fase es concluida con la aceptación de las variables ambientales y con todos los requisitos cumplidos. En este caso, el modelo estará listo para el lanzamiento cuando esté definido y evaluado de acuerdo al alcance establecido en este proyecto. En esta fase concluye el refinamiento del modelo “YoungProgramming”, y la entrega de la documentación asociada.

1.5.5 Fase de Divulgación y Documentación

En esta fase final se hace entrega de la monografía del trabajo de grado incluyendo ahí los resultados obtenidos de la investigación. Por último se realizará el proceso de sustentación del trabajo de grado ante los respectivos jurados de la facultad de Ingeniería Electrónica y Telecomunicaciones.

⁷Sprint: ciclo iterativo donde ciertos temas del modelo (pila de temas del sprint) se realizan para producir un incremento en su definición.

1.6 APORTES

1.6.1 En el ámbito académico

Teniendo en cuenta la necesidad de aprendizaje de programación de manera eficaz para adolescentes que inician su carrera universitaria y su interés en áreas semejantes; es importante y necesario formular modelos que permitan realizar la enseñanza de programación en adolescentes de manera que ellos asimilen este campo de forma fácil, correcta y apropiativa.

Una gran ventaja que ofrece este trabajo es que permite a los jóvenes desarrollar el pensamiento computacional de manera dinámica y agradable, a su propio ritmo y de acuerdo con sus intereses. Además, se promueve el trabajo en equipo y la colaboración, incentivando de esta manera las relaciones interpersonales.

1.6.2 Investigación

Ratificando la importancia que tiene conocer la experiencia de los jóvenes en su interacción con la programación, este trabajo de grado se encamina en la incorporación de las características claves en la colaboración, para que el aprendizaje tenga lugar de manera efectiva en la enseñanza a jóvenes, que forman parte esencial de su experiencia en el área de programación; campo de interés dentro del Departamento de Sistemas de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

1.6.3 Desarrollo

Teniendo en cuenta las necesidades que tiene el sector TIC del país y el alto índice de deserción en el área, este trabajo de grado busca desarrollar el Pensamiento Computacional de manera eficaz para adolescentes que inician su carrera universitaria y su interés en áreas semejantes, logrando así un aporte en la educación y el sector tecnológico del país.

1.7 ORGANIZACIÓN DEL DOCUMENTO

El presente documento se encuentra organizado por capítulos de la siguiente manera:

Inicialmente se presenta una introducción al trabajo, se describe la motivación, se contextualiza el problema, se define la propuesta, los objetivos, la metodología a seguir y los aportes de esta investigación en diferentes ámbitos.

En el segundo capítulo se presenta el marco teórico, mostrando algunas bases conceptuales sobre enseñanza y aprendizaje de la programación, pensamiento computacional, la lúdica, trabajo colaborativo, ChildProgramming y la herramienta Scratch, posteriormente en el capítulo tres se define se describe el método de aplicación de la lúdica y la colaboración para el proceso de enseñanza y aprendizaje de la programación.

En el capítulo cuarto se detalla el modelo YoungProgramming, se explica la reformulación de las prácticas, roles y artefactos, del modelo ya existente, considerando aspectos claves de lúdica y colaboración que faciliten el aprendizaje de la construcción de software en equipos de desarrollo jóvenes, definiendo una nueva arquitectura, un proceso y definiendo la herramientas necesarias para la consecución este trabajo.

En el capítulo quinto se presentan los estudios de caso. El primer caso hace parte una fase exploratoria para evaluar la viabilidad de la investigación, en los casos posteriores se

realizan los estudios de caso, en los que se hace un análisis de la lúdica y colaboración en YoungProgramming.

Finalmente, en el capítulo seis, se presentan el resumen del proyecto, las conclusiones, recomendaciones y los trabajos futuros.

Capítulo 2

2.1 REVISIÓN DE LA LITERATURA Y TRABAJOS RELACIONADOS

Este capítulo contiene una base conceptual (ver **Figura 1**) acerca de lo que es enseñanza y aprendizaje de la programación, pensamiento computacional, lúdica, trabajo colaborativo, además se brinda una explicación de ChildProgramming y la herramienta Scratch [26], también se define el método que se va a utilizar como guía para llevar a cabo el proceso y finalmente los trabajos relacionados en el ámbito de la enseñanza y aprendizaje de la programación.

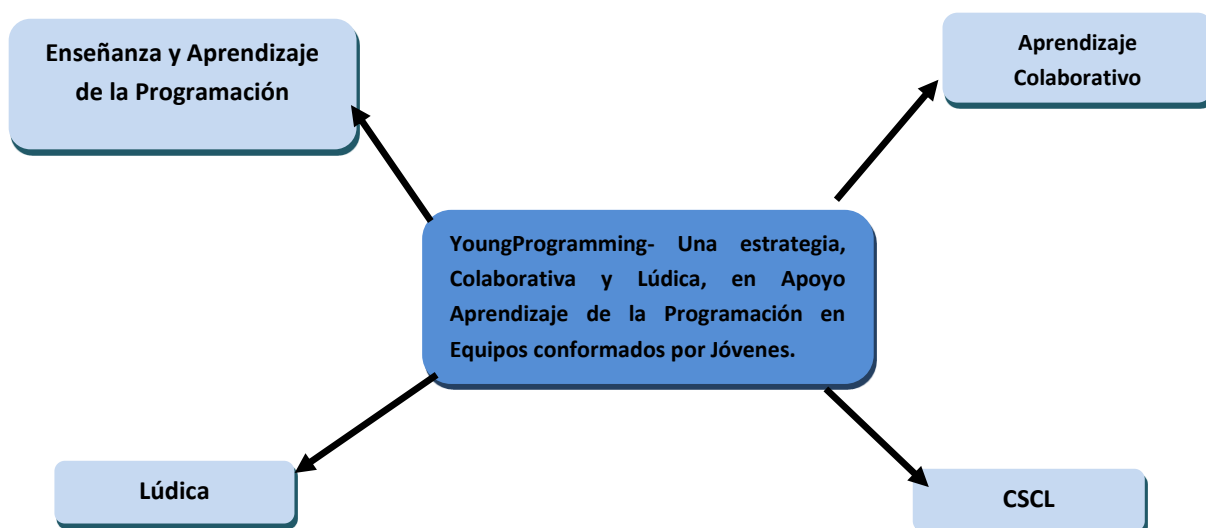


Figura 1. Modelo Conceptual de la Revisión de la Literatura

2.1 Aprendizaje Colaborativo (AC)

El aprendizaje colaborativo se basa en supuestos epistemológicos diferentes y tiene su origen en el constructivismo social [30]. Recoge la esencia de los fundamentos filosóficos del aprendizaje colaborativo: “El aprendizaje colaborativo se produce cuando los alumnos y los profesores trabajan juntos *para crear el saber...* Es una pedagogía que parte de la base de que las personas crean significados juntas y que el proceso las enriquece y las hace crecer” [31].

En vez de dar por supuesto que el saber existe en algún lugar de la realidad “exterior” y que espera ser descubierto mediante el esfuerzo humano, el aprendizaje colaborativo, en su definición más estricta, parte de la base de que el saber se produce socialmente por consenso entre compañeros inmiscuidos en el tema en cuestión. El saber es “algo que construyen las personas hablando entre ellas y poniéndose de acuerdo”[32], este autor es el más decidido defensor del aprendizaje colaborativo, él quiere evitar que los estudiantes se hagan dependientes del profesor como autoridad en los contenidos de la asignatura o en los procesos grupales. Por tanto, en su definición de aprendizaje colaborativo, no le corresponde al profesor la supervisión del aprendizaje del grupo, sino que su responsabilidad consiste en convertirse, junto con los alumnos, en miembro de una comunidad que busca el saber.

En un artículo publicado en Change, llamado “Cooperative learning versus Collaborative Learning”, escrito por [32], el autor dice: *“Describir el aprendizaje cooperativo y el colaborativo como complementarios es subestimar ciertas diferencias importantes entre ambos: algunas tareas que la pedagogía del aprendizaje colaborativo recomienda hacer a los profesores tienden a debilitar parte de lo que espera lograr el aprendizaje cooperativo y viceversa”*. La esencia de su postura es que, mientras que la meta del aprendizaje cooperativo es trabajar juntos en armonía y apoyo mutuo para hallar la solución, la meta del aprendizaje colaborativo es desarrollar a personas reflexivas, autónomas y elocuentes, aunque a veces, esa meta promueve un desacuerdo y una competición que parecen ir en contra de los ideales del aprendizaje cooperativo. Añade que, mientras que la educación cooperativa puede ser apropiada para niños, el aprendizaje colaborativo es más adecuado para los estudiantes universitarios.

Para efectos del estudio de este trabajo de investigación se ha convenido el uso del concepto Aprendizaje colaborativo **(AC)**, entendido como el hecho de que dos o más estudiantes trabajen juntos, compartan equitativamente la carga de trabajo y recursos mientras progresan hacia los resultados de aprendizaje previstos, asuman la responsabilidad de su propio aprendizaje y el de los demás integrantes del grupo, todo esto bajo la tutoría de un docente.

En la **Tabla 1** se exponen las principales diferencias entre el aprendizaje colaborativo y el aprendizaje tradicional.

Grupo de Aprendizaje Colaborativo	Grupo de Aprendizaje Tradicional
Responsabilidad Individual	Ninguna responsabilidad individual
Grupos con integrantes Heterogéneos	Grupos con integrantes Homogéneo
Dirección Compartida	Dirección de un Líder
La responsabilidad es compartida	El sujeto es responsable de sí mismo
Importa tanto la tarea como el proceso	Solamente importa el proceso
Se enseñan habilidades sociales	Se asume que los sujetos ya tienen desarrolladas habilidades sociales
El profesor observa e interviene en el grupo	El profesor no hace caso del grupo

Tabla 1 Diferencia entre Grupos Colaborativos y Grupos Tradicionales Tomado de [49]adaptado de Johnson y Smith [33].

2.2 Distinción entre aprendizaje cooperativo y colaborativo

En una discusión detallada de esta distinción, Dillenbourg [3] ha definido ampliamente esta distinción considerando: En la cooperación, los participantes dividen el trabajo, resuelven las tareas de manera individual y luego ensamblan los resultados parciales en el producto final. En la Colaboración, los participantes realizan el trabajo en forma conjunta [3] [7].

Dillenbourg [3] luego se refiere a la definición de colaboración dada por Roschelle & Teasley [6]: Este capítulo presenta un caso de estudio con el fin de ejemplificar el uso del computador como una herramienta cognitiva para el aprendizaje que ocurre socialmente. Hemos investigado una clase importante de la actividad social, la construcción colaborativa del

conocimiento en la resolución de un problema nuevo. La colaboración es un proceso en el que los individuos negocian y comparten significados relevantes a una tarea de resolución de problemas. La colaboración es una actividad coordinada, sincrónica que es el resultado de un intento continuo de construir y mantener una concepción compartida de un problema [6] [7].

Si alguien está investigando sobre aprendizaje, esto resulta en un contraste significativo. En la cooperación, el aprendizaje es realizado por individuos, quienes contribuyen con sus resultados individuales y presentan el conjunto de resultados individuales como el producto grupal. El aprendizaje en los grupos cooperativos es visto como algo que ocurre de manera individual y puede por lo tanto ser estudiado con los métodos y conceptualizaciones tradicionales de la investigación educativa y psicológica [7]. Por el contrario, en la caracterización de la colaboración dada por Roschelle & Teasley [6], el aprendizaje ocurre socialmente como la construcción colaborativa del conocimiento. Por supuesto, los individuos están involucrados en este aprendizaje como miembros del grupo, pero las actividades en las que ellos participan no son de tipo individual sino grupal, como la negociación y el compartir. Los participantes no se van a realizar las tareas individualmente, sino que se mantienen comprometidos con una tarea compartida la cual es construida y mantenida por todo el grupo [6] [7].

2.3 Pensamiento Computacional.

Wing en [11] define al Pensamiento Computacional como los procesos de pensamiento implicados en la formulación de problemas y representación de sus soluciones, de manera que dichas soluciones puedan ser efectivamente ejecutadas por un agente de procesamiento de información (humano, computadora o combinaciones de humanos y computadoras). Wing es la principal promotora del pensamiento computacional y su visión es que el pensamiento computacional, al igual que el español o la aritmética “será una habilidad y una actitud de aplicación universal para todas las personas”. El Pensamiento Computacional tiene como finalidad desarrollar en los individuos el pensamiento crítico en coalición con los conceptos claves de la computación, tales como abstracción, algoritmos, programación, simulación, entre otros. Con la ideología de que las habilidades y los conceptos de la computación sean difundidos a nivel general, y no únicamente para los ingenieros y especialistas en computación [9][11].

2.4 CSCL dentro de la educación.

El Aprendizaje Colaborativo Apoyado por Computador (CSCL) es un área emergente de las ciencias del aprendizaje referente a estudiar como las personas pueden aprender de manera conjunta con la ayuda de los computadores.

Así como algunas formas particulares del aprendizaje, CSCL está altamente relacionada con la educación. Considera todos los niveles de educación formal, desde el kínder hasta postgrados al igual que en la educación informal. Los computadores se han convertido en un elemento muy importante en este tipo de educación, dado que ya hay políticas gubernamentales alrededor del mundo para dar acceso a los estudiantes a este tipo de tecnologías y acceso a Internet. La idea de fomentar a que los estudiantes aprendan a trabajar en conjunto en grupos pequeños ha sido un aspecto muy enfatizado desde las ciencias sociales. Sin embargo, la habilidad para combinar estas dos ideas (apoyo computacional y aprendizaje colaborativo) con el objetivo de fortalecer el aprendizaje requiere un cambio- un cambio que CSCL se espera lo realice [7].

2.5 Enseñanza de la Programación.

Desde el punto de vista educativo la programación de computadoras posibilita activar una amplia variedad de estilos de aprendizaje, desarrollar el pensamiento algorítmico, y en muchos casos la creatividad e innovación. Posibilita a los estudiantes encontrar diversas maneras de abordar problemas y plantear soluciones, al tiempo que desarrollan habilidades para: visualizar caminos de razonamiento divergentes, anticipar errores, y evaluar rápidamente diferentes escenarios mentales. Los problemas del tipo pensamiento lateral permiten que se exploren y consideren la mayor cantidad de alternativas para solucionar un problema, favoreciendo además que el estudiante explore, escuche y acepte, en algunos casos, diferentes puntos de vista. En las situaciones donde la resolución de un problema involucre búsqueda de información, será necesario e indispensable ayudar a los estudiantes a desarrollar la Competencia para Manejar Información [22]. El estudiante debe poner en práctica habilidades, conocimientos y actitudes, para identificar lo que necesita saber sobre un tema específico en un momento dado, realizar búsquedas efectivas de la información requerida, además de poder determinar si es pertinente al mismo [21][22].

2.6 Lúdica Como Estrategia Didáctica.

El aprender y lo lúdico se presentan, como una dualidad vital diaria y aportan la posibilidad de producir cambios sociales positivos, un espacio de construcción cultural, de conocimiento y lo más importante de encuentro social. La lúdica se toma entonces como una forma de ser, una manera de interactuar con diversas facetas, para hacerlas más manejables en la incertidumbre de la realidad, característica esencial de la vida, del juego y del accionar lúdico.

Las actividades lúdicas mejoran la motivación, atención, concentración, potencia la adquisición de información y el aprendizaje generando nuevos conocimientos. En su accionar vivencial y por su alta interacción con otros y con el medio aumenta la capacidad al cambio, de recordar y de relacionarse dentro de ambientes posibilitantes, flexibles y fluidos. Es importante dar a la lúdica su relevancia en los procesos vitales, formativos y establecer su relación con el juego, el cual ha sido colocado casi como un sinónimo, lo que dista de su acepción más acertada: “manifestación externa del impulso lúdico”, que impide ver otros aspectos por desarrollar de manera amplia en el quehacer lúdico [17].

2.7 Efectos del Juego en la Enseñanza

El juego es una acción que se desarrolla dentro de ciertos límites de lugar, de tiempo, y de voluntad, siguiendo ciertas reglas libremente consentidas, y por fuera de lo que podría considerarse como de una utilidad o necesidad inmediata. Durante el juego reina el entusiasmo y la emotividad, ya sea que se trate de una simple fiesta, de un momento de diversión, o de una instancia más orientada a la competencia. La acción por momentos se acompaña de tensión, aunque también conlleva alegría y distensión [65].

Para Martin Gardner [56] aparecen dificultades al llegar a una definición precisa sobre el juego: la idea de “juego” conlleva muchos significados, enlazados entre si un poco a la manera en que lo están los miembros de una familia humana, significados que han ido concatenándose al tiempo que evolucionaba el lenguaje. Se puede decir que los juegos son educativos –no importa de qué tipo- cargados de una fuerte componente lúdica; aclarando así que las ideas de juego, recreación y lúdico son aproximadamente iguales.

La utilización de juegos en el aula parece que tiene efectos beneficiosos. Al poner en práctica esta actividad es conveniente hacerlo de la manera que resulte más eficaz, en ese sentido,

sea cual sea su nivel de conocimientos el empleo cuidadosamente planificado de juegos puede contribuir a clarificar las ideas del programa y a desarrollar el pensamiento lógico.

Es claro que un educador no puede basar su enseñanza en la exclusiva utilización de juegos. Tampoco se llega a aprender a programar utilizando exclusivamente libros de programación. Lo que parece más conveniente es mantener un equilibrio entre la programación lúdica (que mantiene el interés) y la programación seria (base científica).

Thomas Butler en [57] señala algunos efectos de los juegos en la enseñanza. Algunos resultados de interés son los siguientes:

- Generalmente los estudiantes adquieren por lo menos iguales conocimientos y destrezas que las que obtendrían en otras situaciones de aprendizaje.
- La información es aprendida más deprisa que en otras metodologías, aunque la cantidad aprendida no es significativamente mayor que con otros métodos.
- La resolución del problema conlleva el uso de enseñanza de alto nivel taxonómico. La utilización de juegos, junto a otros recursos, proporcionaría de forma satisfactoria una preparación para la resolución de problemas, aunque falta determinar si este alto nivel es recordado con el paso del tiempo.
- Los estudiantes estarán motivados para participar en la actividad, pero su interés por la materia puede que no se mejore.
- Los juegos y simulaciones producen en los estudiantes una tendencia creciente a asistir regularmente a la escuela.
- Los juegos fomentan los procesos de socialización, incluyendo el fomento de amistades interraciales y de grupos no cohesionados.
- Los juegos han de utilizarse relativamente cercanos al momento del aprendizaje, sobre todo si el juego corresponde a un nivel taxonómico alto.
- La utilización de la fantasía, el estímulo o la curiosidad puede incrementar la efectividad de los juegos.
- Algunos resultados observados al utilizar juegos con alumnos de bajo rendimiento:
 - Los alumnos de pequeña capacidad académica mejoran con frecuencia el rendimiento a causa de un mayor interés.
 - Los estudiantes aprenden habilidades y conceptos tan bien o mejor que alumnos que siguieron las actividades convencionales de lápiz y papel.
 - Los juegos que requieren la participación de varios jugadores en cada juego parecen ser más efectivos que aquéllos que permiten algunos estudiantes simplemente como observadores.
 - Algunos juegos particulares pueden ser más productivos que otros con estudiantes particulares.
 - Una combinación de actividades, implicando tanto juegos como trabajos de papel y lápiz, debería ser el más beneficioso.

Para facilitar el análisis de las diversas aportaciones del juego al desarrollo intelectual, imaginativo, afectivo social de las personas, presentamos una tabla (ver **Tabla 2**) en la que si bien aparece cada aspecto por separado, es importante señalar que el juego nunca afecta a un solo aspecto de la personalidad humana sino a todos en conjunto, y es esta interacción una de sus manifestaciones más enriquecedoras y que más potencia el desarrollo del hombre.

ASPECTOS QUE MEJORA EL JUEGO			
Desarrollo psicomotor	Desarrollo cognitivo	Desarrollo social	Desarrollo emocional
<ul style="list-style-type: none"> - Coordinación motriz - Equilibrio - Fuerza - Manipulación de objetos - Dominio de los sentidos - Discriminación sensorial - Coordinación viso-motora - Capacidad de imitación 	<ul style="list-style-type: none"> - Estimula la atención, la memoria, la imaginación, la creatividad, la discriminación de la fantasía y la realidad, y el pensamiento científico y matemático - Desarrolla el rendimiento la comunicación y el lenguaje, y el pensamiento abstracto 	<p><u>Juegos simbólicos</u></p> <ul style="list-style-type: none"> - Procesos de comunicación y cooperación con los demás - Conocimiento del mundo del adulto - Preparación para la vida laboral - Estimulación del desarrollo moral <p><u>Juegos cooperativos</u></p> <ul style="list-style-type: none"> - Favorecen la comunicación, la unión y la confianza en sí mismos - Potencia el desarrollo de las conductas pro-sociales - Disminuye las conductas agresivas y pasivas - Facilita la aceptación interracial 	<ul style="list-style-type: none"> - Desarrolla la subjetividad del niño - Produce satisfacción emocional - Controla la ansiedad - Controla la expresión simbólica de la agresividad - Facilita la resolución de conflictos - Facilita patrones de identificación sexual

Tabla 2 Aspectos que mejora el juego. Fuente: [59]

2.7.1 Niveles de aplicación de los juegos

De [60] se pueden extraer los siguientes elementos que nos permiten su aplicación en el contexto de la enseñanza de la ingeniería del software a través de juegos:

Aunque no se pretende dar una clasificación de juegos, en este trabajo es útil distinguir los juegos por dos características diferenciadas según [61]:

1. Hay juegos cuya práctica exige a los jugadores que utilicen conceptos o algoritmos incluidos en el desarrollo de estos. Así, un jugador consume su tiempo encontrando la solución del problema propuesto. Es por ello, que esos juegos se denominan juegos de conocimiento. Existen publicados o comercializados muchos juegos de este tipo y su utilización puede efectuarse en diferentes etapas de aprendizaje. Se distinguen tres niveles de aplicación de este tipo de juegos:

- PRE-INSTRUCCIONAL. A través de estos juegos el alumno puede llegar a descubrir un concepto o a establecer la justificación de un algoritmo. De este modo, el juego es el único vehículo para el aprendizaje.
- CO-INSTRUCCIONAL. El juego puede ser una más de las diferentes actividades que el profesor utiliza para la enseñanza de un bloque temático. En este caso, el juego acompaña a otros recursos del aprendizaje.
- POST-INSTRUCCIONAL. Los alumnos ya han recibido enseñanza sobre un tema, y mediante el juego se hacen actividades para reforzar lo que han aprendido. Por lo tanto, el juego sirve para consolidar el aprendizaje.

2. Hay otros juegos cuya práctica exige poner en práctica habilidades, razonamientos o destrezas directamente relacionadas con el modo en el que habitualmente se procede a resolver problemas de programación. Por ello, se llaman juegos de estrategia. Hay unos que son personales o solitarios y en los que el jugador tiene que encontrar la forma de resolverlo; otros son multi-personales, y en los que la tarea consiste en descubrir la existencia de una estrategia que le permita ganar siempre a sus oponentes. Este tipo de juegos es, sin duda, el que más interés ha despertado en los alumnos.

Desde el punto de vista de la enseñanza señalemos que la búsqueda de soluciones de juegos sirve para uno o más de los siguientes objetivos:

- Utilizar diferentes técnicas heurísticas, que ayudaran a la resolución de problemas.
- Potenciar actitudes como las de auto-confianza, auto-disciplina o perseverancia en la búsqueda de soluciones.
- Desarrollar habilidades como las de observación y comunicación.

Para poner en práctica los juegos con los alumnos se hacen una serie de recomendaciones sobre el modo que hay que proceder a lo largo de las experimentaciones según Gairín Sallán [59]:

- Es necesaria la practica el juego antes de presentarlo a los alumnos. Y ello por razones como las siguientes:
 - La eficacia de una actividad depende, en buena medida, del entusiasmo con que la realice el profesor o instructor. Y, por tanto, si un juego agrada personalmente a un profesor, este lo presentara de manera que sus alumnos también disfruten.
 - El profesor podrá observar, y corregir si fuese necesario, aspectos tales como si hay lagunas o errores en las reglas, si hay jugadas que tienen dificultades, si el juego puede llevar a situaciones monótonas, si la duración es excesiva.
 - De su propia experiencia sacará la información sobre los procesos que llevan a la solución, sobre las posibles 'vías muertas', los bloqueos que se puedan producir. Así tendrá más posibilidades de prestar ayuda a sus alumnos en el momento oportuno y en el modo más efectivo.
- El juego hay que proponerlo a los alumnos en el momento preciso. Es decir, que hay que determinar si el juego corresponde al nivel pre, co o post-instruccional. Y en el Último caso hay que practicarlo relativamente próximo al momento en que se introdujo la instrucción.

- El juego ha de utilizarse para el fin adecuado. Los alumnos deben conocer que el juego sirve para potenciar su aprendizaje y, por tanto, no ha de practicarse para cubrir tiempos muertos. Además, el profesor distingue si hay que emplear un juego de conocimiento o de estrategia, buscando, en cada caso, que se adapte a los objetivos educativos previstos.
- El juego hay que practicarlo en la forma correcta. Antes de iniciar el juego hay que dedicar alguna sesión para que los alumnos conozcan el material, para que se entiendan a fondo las reglas, para que todos conozcan la forma en que se gana o se pierde. Después es bueno que se comience practicando alguna jugada sencilla; o que se utilicen reglas sencillas, o que se hagan jugadas para practicar, etc.
- Todos los alumnos han de participar en el juego, que sea una actividad igual para todos, aunque en el desarrollo del mismo pueda haber grupos de alumnos que lo practiquen con diferentes grados de dificultad.
- Se puede recurrir a los juegos comercializados o publicados para ponerlos a los alumnos. Para hacer la elección más conveniente, se debe hacer preguntas como las siguientes: ¿sirve el juego para los objetivos propuestos?, ¿qué conocimientos necesita el alumno para practicar el juego?, ¿qué habilidades se requieren para practicar el juego?, ¿el juego es atractivo teniendo en cuenta la edad y maduración de los alumnos?, ¿los alumnos tienen limitaciones físicas para practicar el juego?, ¿hay problemas de costos o de espacio para la práctica del juego?, ¿existe algún compañero que haya experimentado el juego en situaciones similares a las de mis alumnos? ...

2.7.2 Elementos de los Juegos

Herranz [57] señala que dentro de la gamificación intervienen tres elementos fundamentales: las dinámicas, las mecánicas y los componentes del juego.

Organizándolas de más a menos general:

- **Dinámicas:**

Son aspectos globales a los que un sistema gamificado debe orientarse, está relacionado con los efectos, motivaciones y deseos que se pretenden generar en el participante [57]. Existen varios tipos de dinámicas, entre las cuales destacan:

- *Restricciones* del juego, la posibilidad de resolver un problema en un entorno limitado.
- *Emociones* como la curiosidad y la competitividad que surgen al enfrentarse a un reto [62].
 - *Narrativa* o guion del juego, que permitirá dar una idea general del reto al participante.
 - *Progresión* del juego, es importante que haya una evolución, una sensación de avance en el reto y en el juego. Es importante que el jugador sienta que mejora en el juego.
- *Estatus*, las personas necesitan ser reconocidas.
- *Relaciones* entre los participantes.

- **Mecánicas**

Son una serie de reglas que intentan generar juegos que se puedan disfrutar, que generen una cierta “adicción” y compromiso por parte de los usuarios, al aportarles retos y un camino por el que transitar, ya sea en un videojuego, o en cualquier tipo de aplicación [63].

Existen varios tipos de mecánicas de juego [57] :

- *Retos*, sacando a los usuarios de un ambiente de confort para introducirlos en la mecánica del juego [64]. Ojo con no frustrar a los participantes.
- *Oportunidades*, competición y colaboración, planteando la forma idónea en la que comportarse en el juego los participantes.
- Dentro de las oportunidades el jugador tendrá diferentes *turnos*, distintas formas de interactuar contra el juego o contra otros participantes.
- Se podrán dar también asociaciones entre jugadores a modo de *cooperación o formar equipos* para cumplir el reto o la meta.
- La superación de retos u obstáculos irá dando puntos a los participantes.
- Otros elementos serán la *clasificación*, de participantes en función de sus puntos, y la definición de niveles.
- *Realimentación o feedback*, indicará el hecho de obtener premios por acciones bien realizadas o completadas.
- Es importante que el participante se sienta reconocido y para ello se establecen *recompensas*, las cuales pueden ser escalonadas en función al esfuerzo, nivel, riesgo, entre otros [57].

- Componentes

Elementos concretos o instancias específicas asociadas a los dos anteriores. Pueden variar de tipo y de cantidad, todo depende de la creatividad en que se desarrolle el juego.

Destacan:

- *Logros*, regalos, conquistas y/o avances, es importante que se satisfagan una o más necesidades de los participantes.
- Avatares
- Insignias
- “Malos” de final de nivel (Boss Fights)
- Combates
- Desbloques: contenido bloqueado, para abrir...
- Niveles
- La formación de *equipos* motiva la socialización y la sensibilización de las personas en unirse para competir, lograr objetivo común, obtener una recompensa final, etc.
- Puntos
- Tablas de clasificación
- Pruebas
- Objetos virtuales

2.8 Actividad Colaborativa

Para Johnson & Johnson en [34] una actividad colaborativa consiste en el desarrollo de una tarea en grupo con un único objetivo final, intercambiando ideas y materiales, una subdivisión de tareas y recompensas grupales. En definitiva, alumnos trabajando en grupo que intercambian ideas, se hacen preguntas, todos escuchan y comprenden las respuestas, se ayudan entre ellos antes de pedir ayuda al tutor.

El aprendizaje colaborativo eficaz se basa en la argumentación y en el conocimiento compartido. Todas las teorías de enseñanza destacan la importancia de que los estudiantes reflexionen sobre cómo llegaron al resultado final [35]. En el enfoque colaborativo es objeto de interés tanto la solución como el proceso que permite al grupo llegar a ella. Aspectos significativos del proceso, pueden ser representados explícitamente [36]. Este meta nivel juega un papel muy importante en una actividad colaborativa, ya que permite tanto a los alumnos como al profesor, analizar la forma de trabajar de cada grupo. Además, es una valiosa fuente empírica para deducir mecanismos de intervención pedagógica adecuados a cada grupo.

Johnson & Johnson en [34] señalan algunos efectos de la cooperación en las actividades Colaborativas:

- Mayores esfuerzos por lograr un buen desempeño (Mas rendimiento y más productividad, más motivación, más tiempo dedicado a la tarea, mejor retención a largo plazo, niveles superiores de pensamiento crítico y razonamiento).
- Mejores relaciones entre alumnos (espíritu de equipo, solidaridad, compromiso, valoración de la diversidad, cohesión, solidaridad).
- En las actividades colaborativas se procura que los alumnos utilicen conocimientos y estrategias que ya poseen.
- Ayuda a los alumnos a clarificar su pensamiento y desarrollar este a niveles superiores.
- Ayuda a los alumnos a reflexionar y evaluar su propio aprendizaje.
- El tutor interviene adecuadamente, asumiendo el rol de tutor.

El éxito de una persona está relacionado con el éxito de los demás en actividades de aprendizaje colaborativo. Este aspecto es conocido como la interdependencia positiva.

2.8.1 Características de una actividad colaborativa

2.8.1.1 Interdependencia positiva

La interdependencia positiva es la base central del aprendizaje colaborativo, para [37] el aprendizaje no aumenta solo con el hecho de reunir a los alumnos y permitir su interacción, puesto que los alumnos pueden facilitar u obstruir el aprendizaje de los demás o ignorar a sus demás compañeros. La interdependencia positiva hace que los alumnos se preocupen por estimular el aprendizaje y el logro de sus compañeros. Para Johnson & Johnson [37] cuando la interdependencia positiva se comprende claramente, se hace evidente que el esfuerzo de cada uno de los integrantes resulta indispensable para el éxito del grupo y que cada integrante tiene un aporte personal y único para hacer al esfuerzo conjunto, por sus propios recursos o por su rol y las responsabilidades de la tarea. En [38] la interdependencia positiva es el mecanismo que logra e incentiva la colaboración dentro de los grupos de trabajo, animando el grupo a lograr sus objetivos y maximizando de esta forma el aprendizaje

individual, ya que si no se genera interacción entre los integrantes del grupo no hay comunicación y por ende el aprendizaje va a ser más difícil.

En [39], [33] se menciona que la existencia de interdependencia sólidamente estructurada, requiere del esfuerzo de cada integrante del grupo, donde cada integrante del grupo debe tener una contribución única debido a su rol y responsabilidad dentro del grupo. Con lo anterior se crea compromiso con el éxito del grupo en cada integrante, y este es el núcleo del aprendizaje colaborativo. Si no hay una interdependencia positiva no hay colaboración [34].

Tipos de interdependencia positiva [34].

- **Interdependencia positiva de objetivos**

Cada integrante del grupo debe preocuparse por aprender uno del otro, puesto que todo el grupo está motivado para lograr un objetivo común, de manera que los estudiantes deban percibir que pueden lograr su aprendizaje si y solo si los otros integrantes del grupo alcanzan sus objetivos.

- **Interdependencia positiva de recompensas**

Cuando el grupo logre una meta u objetivo las recompensas deben ser conjuntas, de modo que cada integrante del grupo perciba una misma recompensa. Los maestros deben dar a los estudiantes una calificación de grupo si todos los estudiantes lograron los objetivos establecidos, para así incentivar la cooperación en el grupo.

- **Interdependencia positiva de recursos.**

Cada integrante del grupo tiene solo una parte de la información, recursos o materiales necesarios para realizar una tarea, de manera que para poder lograr las metas y objetivos debe ser necesario combinar todos los recursos entre todo el grupo. Los docentes deben de fomentar en los estudiantes la importancia de compartir estos recursos para que el grupo tenga éxito.

- **Interdependencia positiva de roles**

Cada integrante del grupo tiene roles que están interconectados y que le dan responsabilidades específicas para que el grupo en conjunto logre completar una tarea. Los docentes deben asignar a cada estudiante funciones complementarias para lograr una alta calidad del aprendizaje.

- **Interdependencia positiva de identidad**

Cada integrante del grupo tiene que encontrar y acordar una identidad, que puede ser un nombre, un lema, un eslogan, una bandera o una canción.

- **Interdependencia positiva de medio ambiente**

Los integrantes del grupo están unidos por el medio ambiente en el que trabajan. Por lo que los docentes deben encontrar ya sea física o virtualmente un entorno en el que los estudiantes puedan resolver una situación problemática en el desarrollo de una actividad de colaboración.

- **Interdependencia positiva de la fantasía**

El profesor da a los estudiantes una tarea imaginaria, para la cual los estudiantes deben encontrar soluciones para situaciones extremas, como la vida en peligro o manejo de una tecnología en el futuro.

- **Interdependencia positiva de tareas**

El trabajo tiene que estar organizado de manera secuencial. Los integrantes del grupo deben dividir el trabajo y estar vinculados entre sí. Tan pronto como un equipo lleve a cabo su parte de la tarea, el siguiente equipo puede continuar con su responsabilidad, continuando con el ciclo hasta terminar la tarea.

- **Interdependencia positiva respecto del enemigo exterior**

En este tipo de interdependencia, los docentes ponen los grupos en competencia unos a otros. Y de esta manera cada integrante del grupo se siente interdependiente y hace lo mejor para ganar la competencia y superar los resultados de los demás grupos[40].

2.8.1.2 Responsabilidad individual

Para Johnson & Johnson [37] hay responsabilidad individual en un grupo cuando se evalúa el desempeño de cada integrante y se devuelven los resultados tanto a él como a su grupo, de manera que cada uno sea responsable ante sus compañeros por aportar su parte del éxito al grupo. La responsabilidad individual es la clave para asegurar que todos los integrantes de un grupo se ven forzados por el aprendizaje colaborativo. Esta hace que los integrantes de un grupo sepan que no pueden escudarse en el trabajo de los demás y que también queden mejor preparados para realizar actividades cooperativas por sí mismos. Sapon [40] denomina a este componente “*la rendición de cuentas personal*”, pero aclara que aunque es responsable del aprendizaje, no debe esperarse que todos los estudiantes aprendan lo mismo, y en este sentido es que puede darse expresión a las diferencias personales y a las necesidades educativas de cada quien.

En [37] se mencionan algunas formas comunes de estructurar la responsabilidad individual:

1. Formar grupos pequeños
2. Tomar pruebas individuales
3. Tomar evaluaciones orales al azar
4. Observar a cada grupo y registrar la frecuencia de participación de cada integrante.
5. Asignar a un alumno de cada grupo el papel de verificador. El verificador tiene la misión de pedir a sus compañeros que expliquen el razonamiento y las ideas subyacentes en las respuestas grupales
6. Pedir a los alumnos que le enseñen a otro lo que han aprendido. Cuando todos los alumnos hacen esto, se lo denomina explicación simultánea.

2.8.1.3 Igual Participación

En [41] esta característica se refiere a la capacidad de dominar y ejecutar la parte del trabajo de la cual el alumno se ha responsabilizado (o lo han responsabilizado) dentro de un grupo de aprendizaje cooperativo. Para un verdadero trabajo colaborativo, cada miembro del grupo debe ser capaz de asumir íntegramente su tarea y además debe tener los espacios para que pueda participar y contribuir individualmente. En [42] cada miembro del grupo debe asumir íntegramente su tarea y además, tener los espacios para compartirla con el grupo y recibir sus contribuciones

2.9 Diseño de Procesos Colaborativos

A continuación se presenta la definición de los elementos del proceso colaborativo según [43].

2.9.1 Actividad

Una actividad se define como un conjunto de tareas que pueden combinar tanto trabajo individual como de grupo. Las tareas individuales son realizadas por cada alumno, un ejemplo de ellas son tareas de lectura y entendimiento de un tema específico a partir de un material de trabajo. Las tareas de grupo se realizan mediante un proceso de discusión y elaboración común en el que hay un intercambio de ideas y contribuciones para construir progresivamente una solución [43][44].

2.9.2 Roles

Este concepto está vinculado a la función o papel que cumple cada integrante del grupo dentro de la actividad de aprendizaje colaborativo. Esta es una forma efectiva de asignar responsabilidad a todos y cada uno de los integrantes del salón de clases [43][44].

2.9.3 Recursos o Materiales

Con respecto a los materiales el docente debe decidir que materiales serán necesarios para que los alumnos trabajen de manera colaborativa. Los materiales que se requieren para una actividad colaborativa son los mismos que se requieren para otras actividades curriculares, pero hay ciertas variaciones en el modo de distribuir esos materiales que pueden incrementar la cooperación entre los alumnos [34].

2.9.4 Grupos

En el aprendizaje colaborativo no existe una dimensión ideal para los grupos. La dimensión de estos dependerá de los objetivos de las actividades, de las edades de los alumnos y su experiencia en el trabajo en equipo, de los materiales y equipos a utilizar y del tiempo disponible para la clase. Los grupos de aprendizaje cooperativo suelen tener de dos a cuatro miembros. La regla empírica a aplicar es: “cuanto más pequeño sea el grupo, tanto mejor” [34].

2.9.5 Objetivos

Con respecto a los objetivos el docente debe establecer, tanto objetivos específicos como generales junto al grupo de estudiantes, estos objetivos deben enmarcar, objetivos procedimentales relacionados con la adquisición de habilidades, y, objetivos actitudinales relacionados con las competencias motivacionales y emocionales. El objetivo debe estar redactado de forma que inmiscuya a todo el equipo de trabajo y este se sienta comprometido con el desarrollo y logro del mismo.

2.9.6 Reglas de juego

Consiste en definir reglas y especificaciones de cómo y cuándo se realizará la actividad de aprendizaje colaborativo según el criterio del docente, estas reglas se relacionan con fechas, condiciones iniciales de trabajo. Las reglas particulares dependen, por supuesto, del contexto del salón de clase. Por ejemplo, algunas se establecen para dar igual oportunidad

de participación, valorar las opiniones de los demás y argumentar en contra de las ideas de los otros.

2.9.7 Criterios de éxito

Consiste en una breve descripción de los comportamientos esperados, la lista de productos deseados y las metas de aprendizaje deseados.

2.10 Modelado de Procesos Colaborativos

2.10.1 CIAM una propuesta para modelar procesos colaborativos

CIAM (Collaborative Interactive Applications Methodology) [54] es una propuesta metodológica para el diseño y modelado de sistemas interactivos de trabajo en grupo. Esta propuesta implica la adopción de distintos puntos de vista a la hora de abordar la creación de modelos conceptuales para este tipo de sistemas. Las primeras etapas abordan un modelado centrado en el grupo, pasando en fases posteriores a un modelo más centrado en el proceso (cooperativo, colaborativo y de coordinación), acercándonos, a medida que bajamos en el nivel de abstracción, hacia un modelado centrado en el usuario, en el que se modelan las tareas interactivas, esto es, el diálogo que se da entre un usuario individual y la aplicación. Los dos primeros enfoques de modelado permiten la definición del contexto en el que se creará el modelo interactivo, y sirven de punto de partida para este último. La información especificada en cada una de las fases sirve de base para el modelado a realizar en la fase siguiente; de forma que dicha información se amplía, se relaciona o se especifica con un mayor nivel de detalle en la siguiente etapa del proceso.

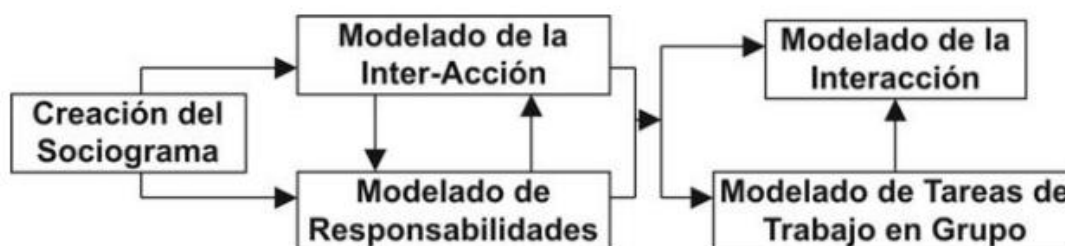


Figura 2. Etapas de la metodología CIAM. Fuente: [54].

Las etapas de las que consta esta propuesta (ver **Figura 2**), y el objetivo de cada una de ellas se enumeran a continuación:

- 1. Creación del Sociograma.** En esta fase se modela la estructura de la organización, así como las relaciones que existen entre los distintos integrantes de la misma. Los integrantes que forman la organización podrán entrar en una de las siguientes categorías: roles, actores, agentes software; o agrupaciones de los anteriores, dando lugar a grupos o equipos de trabajo. Los elementos de estos diagramas podrán relacionarse mediante tres tipos de relaciones básicas (herencia, desempeño y asociación).
- 2. Modelado de Responsabilidades.** En esta fase es de gran ayuda la creación de la llamada tabla de participación. El empleo de esta herramienta de especificación, de naturaleza textual, permite al diseñador tener una primera idea de la división del trabajo al nivel más elevado de abstracción. Una vez construida la tabla de

participación nos centramos en la definición del Modelo de Responsabilidades. La información expresada mediante las técnicas anteriores sirve de base para la definición del modelo de responsabilidades asociado a cada uno de los roles del sistema. Haciendo una lectura por columnas (por roles) de la tabla anterior, completamos las tareas que tiene que realizar cada rol, añadiendo aquellas que son de carácter individual y no están envueltas en los procesos de trabajo en grupo de la organización. Para cada objeto se indican los modificadores de acceso al mismo (**C**, indica creación; **L**, lectura y **E**, escritura).

3. **Modelado de la Inter-Acción.** Entendida ésta como la interacción/colaboración que se produce entre los miembros del grupo. Este modelo permite especificar el funcionamiento completo del proceso de grupo que puede ser cooperativo, colaborativo o mixto. Se define mediante un diagrama de estados que permitirá relacionar toda la información definida mediante las dos técnicas anteriores. Este diagrama se representa mediante un grafo cuyos nodos son las actividades que componen el trabajo en grupo y cuyos arcos indican relaciones entre dichas actividades (de orden, de dependencia de datos, de condición, notificación, de paso del tiempo, etc.). Cada nodo indica el nombre de la tarea a realizar, su tipo, los roles envueltos en su ejecución y los objetos manejados en la misma.
4. **Modelado de Tareas de Trabajo en Grupo.** En esta fase se definen con un mayor nivel de detalle las tareas en grupo identificadas en la etapa anterior.
5. **Modelado de Interacción.** En esta última fase se modelan los aspectos puramente interactivos de la aplicación. Se creará un modelo de interacción para cada tarea de naturaleza individual detectadas en las distintas fases del proceso de refinamiento gradual de modelo.

2.10.2 AMENITIES una metodología para el análisis y diseños de sistemas cooperativos

AMENITIES (A METHodology for aNalysis and desIgn of cooperaTIve systEmS) [55] constituye una propuesta metodológica para el estudio y desarrollo de sistemas cooperativos. Tiene entre sus principales objetivos proporcionar [55]:

- Procesos para estudiar y desarrollar sistemas colaborativos bajo un mismo marco de trabajo conceptual incluyendo los conceptos relevantes (y las relaciones entre ellos) para este tipo de sistemas.
- Una descripción integrada de requisitos obtenidos y modelado del sistema tras aplicar una combinación de técnicas. Se utiliza un único lenguaje con la intención de facilitar la comunicación de estos requisitos. En particular, se utilizan los diagramas de casos de uso de UML para estructurar y describir los requisitos, y una notación, basada también en los diagramas de estados y actividades de UML, para el modelado del sistema.
- Un modelo de sistema que represente todos los conceptos del marco de trabajo conceptual permitiendo la negociación de requisitos y el diseño del nuevo sistema. Con este objetivo se construye el modelo integrando de forma jerárquica un conjunto de modelos de comportamiento y tareas.

- Un marco de trabajo metodológico que permite combinar métodos y técnicas de ingeniería de requisitos (etnografía, casos de usos como agrupación de escenarios, etc.) con el resto de aspectos a cubrir en cuanto al desarrollo (diseño arquitectónico, etc.) desde el punto de la Ingeniería del Software.

AMENITIES permite describir un sistema colaborativo mediante cuatro vistas [55] que facilitan detectar los aspectos más relevantes de este tipo de sistemas.

- **Vista de grupo:** Primeramente, se deben identificar los aspectos relacionados con la propia organización (el grupo), y las restricciones que impone esta asociación. Las organizaciones se articulan bajo el concepto de rol, que determina la relación entre los miembros del grupo y las tareas que deben llevar a cabo. A menudo esta relación está condicionada por una serie de restricciones impuestas al sistema colaborativo, y de las cuales podemos identificar las siguientes como más importantes:
 - *Capacidades.* Esta es una restricción cognitiva que se impone a cada actor para participar con un rol determinado. Estas capacidades determinan los conocimientos que debe adquirir un usuario para participar con un rol concreto.
 - *Leyes.* Este tipo de restricción viene impuesta por la propia organización e identifica las reglas que deben ser preservadas en el grupo. Normalmente estas reglas se deducen de la propia estructura social que se manifiesta en el grupo (democrática, jerárquica, etc.)

Ambas restricciones permiten modelar sistemas dinámicos, ya que es habitual que tanto la estructura del grupo como su funcionamiento se modifique en el tiempo (los participantes pueden adquirir nuevas capacidades, variar en número de miembros que lo conforman o bien, modificar las leyes que rigen el grupo al aplicar nuevas estrategias de trabajo).

- **Vista cognitiva:** La vista cognitiva representa el conocimiento que posee o adquiere cada miembro del grupo en el escenario colaborativo. Este conocimiento queda reflejado mediante la descripción de las tareas que puede llevar a cabo. La descripción de las tareas implica un análisis profundo de las actividades que se deben realizar en el grupo, la división del trabajo y determinar las interrelaciones que existen entre ellas. El análisis de tareas contempla todos estos pasos, si bien lo hemos dividido en dos fases claramente diferenciadas. En primer lugar definimos lo que denominamos interfaz del rol, que recoge las características más relevantes de las tareas a desempeñar por un rol junto a las interrelaciones con el resto de participantes (tareas) y entorno (mediante eventos).
- **Vista de interacción:** Otro aspecto que se deben estudiar son los procesos que implican un diálogo entre participantes para analizar sus características, concretamente:
 - El modo de diálogo que se producen entre participantes
 - Los requisitos que impone ese diálogo sobre los medios a utilizar este modo de diálogo lo identificaremos mediante protocolos.

Los protocolos se pueden analizar por separado dentro de la organización ya que en gran medida son independientes del dominio del problema, y por tanto, se pueden incorporar al análisis de tareas. Por ejemplo, se pueden identificar protocolos democráticos (toma de una decisión por mayoría), consenso (aprobación unánime de una decisión), jerárquica, etc.

- **Vista de información:** Por último, se debe recoger la información que es compartida en el escenario. Esta información se puede describir de manera implícita en las actividades y acciones o bien, de modo explícito como flujo de información entre actividades. La información que fluye a través del sistema colaborativo serán los documentos (los objetos que son gestionados en el sistema), eventos y recursos.

2.11 Mecanismos para fomentar el Aprendizaje Colaborativo

En esta sección se describirán algunos aspectos claves para fomentar la colaboración en el aprendizaje colaborativo, según algunos autores reconocidos:

Según[45], los estudiantes influyen de gran manera en el cómo se desarrolla la actividad colaborativa, de ellos depende el éxito o no de la colaboración. Para él, los estudiantes que estén comprometidos en el proceso de aprendizaje tienen las siguientes características:

- **Responsables por el aprendizaje:** Los estudiantes se hacen cargo de su propio aprendizaje y son auto regulado.
- **Definen los objetivos del aprendizaje y los problemas que son significativos para ellos,** entienden que actividades específicas se relacionan con sus objetivos, y usan estándares de excelencia para evaluar qué tan bien han logrado dichos objetivos.
- **Motivados por el aprendizaje:** Los estudiantes comprometidos encuentran placer y excitación en el aprendizaje. Poseen una pasión para resolver problemas y entender ideas y conceptos. Para estos estudiantes el aprendizaje es intrínsecamente motivante.
- **Colaborativos:** Los estudiantes entienden que el aprendizaje es social. Están “abiertos” a escuchar las ideas de los demás, a articularlas efectivamente, tienen empatía por los demás y tienen una mente abierta para conciliar con ideas contradictorias u opuestas. Tienen la habilidad para identificar las fortalezas de los demás.
- **Estratégicos:** Los estudiantes continuamente desarrollan y refinan el aprendizaje y las estrategias para resolver problemas. Esta capacidad para aprender a aprender (meta cognición) incluye construir modelos mentales efectivos de conocimiento y de recursos, aun cuando los modelos puedan estar basados en información compleja y cambiante.

Según [46], algunos elementos básicos para fomentar el aprendizaje colaborativo son:

- Participación activa de los estudiantes
- Establecer y mantener un entendimiento compartido
- Pensamiento crítico (explicar, justificar, motivar)
- Evaluar el progreso
- Ayuda mutua

Según Johnson & Johnson en [47], los ambientes de aprendizaje colaborativo son los que favorecen de mejor forma el aprendizaje de habilidades intelectuales de orden superior, tales como:

- Resolución de problemas
- Pensamiento crítico
- Pensamiento creativo
- Pensamiento meta cognitivo
- Retención de información

Según [48], los mecanismos que operan en una situación colaborativa son:

- Conflicto o desacuerdo
- La propuesta alternativa
- Auto-explicación
- Apropiación
- Esfuerzo cognitivo compartido
- Regulación mutua

2.12 CHILDPROGRAMMING

Para facilitar la enseñanza de la programación en niños, el grupo IDIS ha propuesto el modelo ChildProgramming, desarrollado por Hurtado et al [1] se recrea un ambiente de desarrollo para niños como estrategia de aprendizaje y construcción de software basada en la lúdica, la colaboración y la agilidad. Según los autores, con este modelo se ofrece un espacio a los niños para desarrollar sus habilidades lógico-matemáticas y sociales, además se les ofrece a los alumnos la libertad, para facilitar que emerjan en este ambiente de desarrollo aportando a la misión de las instituciones educativas. Esta propuesta metodológica utiliza estrategias de trabajo colaborativo y aprovecha los enfoques ágiles para el desarrollo de software con fines de facilidad y de descubrimiento.

El objetivo principal de este modelo es guiar a un equipo pequeño de niños hacia el desarrollo de una solución software efectiva, basándose en el desarrollo cognitivo del niño teniendo en cuenta la teoría de estadios de Piaget [27] en la que los niños entre edades comprendidas entre los 7 y 11 años están en el estadio cognitivo de operaciones completas. El componente colaborativo se basó en la mayor interacción en el aprendizaje para lograr un mejor desarrollo operacional y social que es útil en varias áreas para los niños.

El modelo es el resultado de análisis del comportamiento de los niños en grupo, el cual incluye las 10 mejores prácticas adoptadas por los niños que bajo el estudio resultaron relevantes: los niños entienden conceptos, aceptan reglas, siguen ordenes e instrucciones, tienen voluntad para trabajar en equipo, hacen diseños y soluciones simples, realizan entregables de producto, trabajan en parejas y en equipo, despejan dudas e inquietudes. De estas prácticas se analizó la relación que tienen con el componente colaborativo, cognitivo y ágil, y se propuso un proceso sobre un marco de trabajo en Scrum basado en 3 etapas pre-juego, juego y post-juego como se muestra en la **Figura 3**.

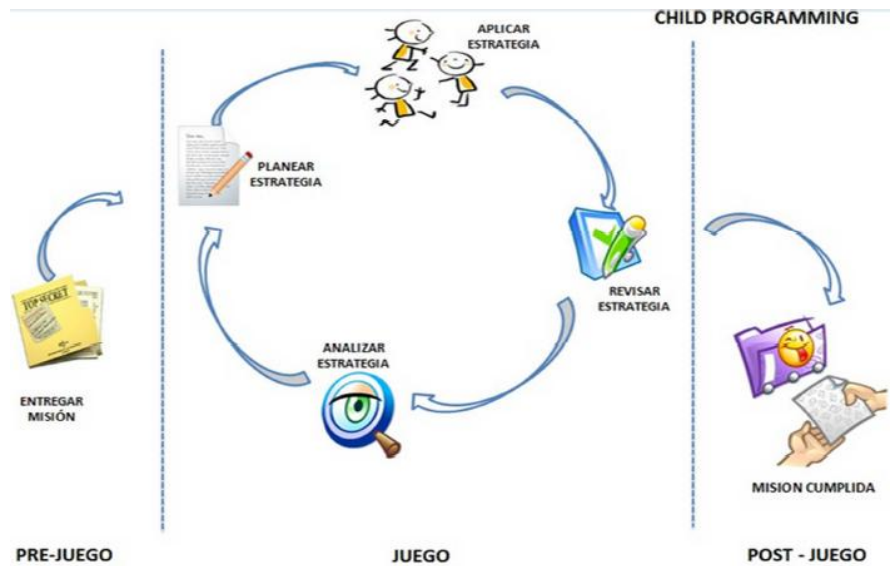


Figura 3. Ciclo de vida de ChildProgramming. Fuente: [1].

También se extrajo un catálogo de Elementos de ChildProgramming, donde se evidenciaron roles (internos y externos), prácticas y conceptos ágiles y colaborativos, ambiente de trabajo, humana, ambiental, para finalizar con un estudio de caso tres en el que se evaluó el modelo, para el desarrollo de esta actividad se trabajó con un grupo experimental y otro grupo de control (que trabajaba de manera intuitiva), en esta actividad computacional, se enfrentaban al lenguaje de programación ya con un esquema de trabajo, en donde tendrían que elaborar piezas funcionales de software, en esta actividad se pidió calcular índice de masa corporal de una persona.

Las conclusiones fueron que a nivel de comportamiento y productividad se vieron resultados positivos donde los niños se apropiaron de la metodología, se apropiaron del proceso y de los conceptos, adoptando las mejores prácticas. Además se desarrolló una prueba piloto donde se aplicó el modelo en ciencias naturales y dio resultados positivos.

Este modelo aporta significativamente en la enseñanza de la construcción de programas orientada a los niños, pero no está definido para jóvenes, por lo tanto es necesario redefinir las prácticas y artefactos de ChildProgramming (véase **Figura 4**) centrándose en la parte lúdica y colaborativa para que los adolescentes aprendan a programar en entorno motivado basado en los aspectos anteriormente mencionados.



Figura 4. Modelo inicial planteado de ChildProgramming. Fuente: [1].

El actual trabajo sigue esta misma línea de investigación, pero reformulando prácticas, roles y artefactos de ChildProgramming, considerando aspectos claves de lúdica y colaboración que faciliten el aprendizaje de la construcción de software en equipos de desarrollo jóvenes.

2.13 SCRATCH

Scratch fue desarrollado por el “Lifelong Kindergarten group” en el Media Lab del Instituto Tecnológico de Massachusetts como un lenguaje de programación visual para niños de 6 años en adelante. Desde que se liberó en 2007, el sitio web de Scratch tiene más de 800,000 usuarios y en él se pueden hallar más de 1.7 millones de diversos proyectos, la mayoría relacionados con juegos y animaciones sencillas. Scratch es una herramienta que fomenta la creatividad de los niños, permitiendo explorar y experimentar con diferentes creaciones[26].

El nombre de Scratch se deriva de la técnica de “scratching” usada en el “Turntablism” (arte del DJ para usar los tornamesas), y se refiere tanto a la lengua y su aplicación. La similitud con el “scratching” musical es la fácil reutilización de piezas: en Scratch todos los objetos, gráficos, sonidos y secuencias de comandos pueden ser fácilmente importados a un nuevo programa y combinados de diferentes maneras permitiendo a los programadores conseguir resultados rápidos y estar motivados para intentar más. Se puede utilizar este programa para, tal y como dice su lema, programar, jugar y crear.

Las razones por las cuales se escogió esta herramienta para el desarrollo de las actividades y los estudios de caso se basan en las dificultades identificadas por Brusilovsky[25], las cuales pueden resumirse en los siguientes puntos:

- Los lenguajes de propósito general son demasiado extensos y poseen demasiadas características particulares. La base conceptual del lenguaje junto a los principios de la programación hace que exista una gran cantidad de material a asimilar por parte del estudiante. En lugar de hacer hincapié en los principios fundamentales, los lenguajes de propósito general se pierden en nociones secundarias y sutilezas concernientes a la implementación.

- Los lenguajes de propósito general proporcionan pocas facilidades para entender las nociones básicas y estructuras de control. Los lenguajes no son visuales y la ejecución está oculta. Con el proceso de ejecución oculto, los estudiantes desarrollan un aprendizaje orientado a entradas y salidas.
- Dado que los lenguajes de propósito general están enfocados a procesar números y símbolos, los primeros problemas que suelen darse en la enseñanza del lenguaje es que se encuentra lejos de la experiencia diaria de los estudiantes y no les resulta atractivo.

Para solucionar estos problemas Brusilovsky propone el uso de mini-lenguajes [25], es decir, lenguajes de programación con una sintaxis reducida y una semántica simple. Por tal razón, se escoge Scratch como herramienta para el desarrollo del proyecto, ya que permite realizar hincapié en los principios fundamentales y no en nociones secundarias y sutilezas concernientes a la implementación.

2.14 Trabajos Relacionados

Hurtado et al. [1] proponen un modelo llamado ChildProgramming (ChP) en el que se recrea un ambiente de desarrollo para niños como estrategia de aprendizaje y construcción de software basada en la lúdica, la colaboración y la agilidad. Según los autores, con este modelo se ofrece un espacio a los niños para desarrollar sus habilidades lógico-matemáticas y sociales, además se les ofrece a los alumnos la libertad, para facilitar que emerjan en este ambiente de desarrollo. Esta propuesta metodológica utiliza mecanismos de aprendizaje, estrategias que conllevan a una verdadera colaboración, haciendo uso de técnicas colaborativas comúnmente utilizadas, y la lúdica. Sin embargo esta última parte de la propuesta aún no ha sido desarrollada y evaluada. Nuestro trabajo sigue esta misma línea de investigación, pero intenta evaluar y aplicar la utilización de una metodología colaborativa y lúdica en jóvenes para la enseñanza de la programación.

Meerbaum-Salant et al. [13] desarrollaron materiales de aprendizaje para estudiantes de secundaria, los cuales fueron diseñados de acuerdo a la filosofía de Scratch. Las pruebas fueron realizadas a jóvenes entre 14 y 15 años. La estrategia utilizada fue enseñar a los estudiantes los conceptos básicos de la programación tales como inicialización, estructuras condicionales, estructuras repetitivas, el paso de mensajes, variables y concurrencia a través de la relación que tienen estos conceptos con conceptos de otras áreas de conocimiento. Al final de las actividades los estudiantes eran interrogados para que dieran la definición de estos conceptos básicos de la programación. La gran mayoría de respuestas eran relacionadas al conocimiento en otras áreas, principalmente de las matemáticas. En conclusión, los autores destacan la importancia de Scratch como una plataforma viable para la enseñanza de la programación, sin embargo se encontraron problemas con tres conceptos: inicialización, variables y concurrencia, ya que son conceptos mucho más abstractos que los otros ya mencionados. Los autores atribuyen esto al diseño del curso y a la poca implicación del profesor como parte del proceso de aprendizaje. El trabajo que proponemos buscará mediante un modelo colaborativo que potencie la comunicación entre los individuos involucrados en el proceso de aprendizaje.

Javidi y Sheybani. [14] desarrollaron un plan para estudiantes de bachillerato en Virginia, USA, llamado DIGINSPIRED, cuyo objetivo principal era relacionar los intereses naturales de los jóvenes en los juegos interactivos y así motivarlos a aprender a programar y explorar la tecnología, además de inculcar la ingeniería de software como carrera.

El trabajo consistió en crear juegos cuya temática abordara una de cuatro áreas de ciencias: reciclaje, nutrición, ejercicio físico o consumo de drogas.

Para ayudar a los estudiantes a diseñar sus juegos, los participantes juegan otros juegos de computador para generar ideas, y así logran tener diferentes opciones para elegir la historia de su juego. A lo largo de la fase de construcción, los participantes presentan sus juegos, reciben retroalimentación sobre cómo podrían mejorarlo, e identifican "bugs" para posteriormente corregirlos. Como conclusión, los autores destacan que con el uso de muchas de las actividades de exploración, los estudiantes han aumentado su comprensión de los procesos involucrados en el diseño del juego y de la importancia de la programación para llevar a cabo ciertas tareas. Sin embargo, los autores subrayan la dificultad que tuvieron los estudiantes para trabajar con sus otros compañeros, ya que los jóvenes preferían hacerlo de manera individual. Nuestro trabajo pretende facilitar la comunicación entre los estudiantes para alcanzar la colaboración necesaria para activar los procesos de aprendizaje en cada uno de ellos.

Jurado et al. [15] en este artículo se expone cómo se ha adaptado un sistema preexistente para el aprendizaje de la Programación basado en Eclipse, de modo que tenga soporte para realizar tareas colaborativas. Estas han sido especialmente diseñadas para la resolución de problemas de Programación, e integradas dentro del entorno de aprendizaje. Los servicios que implantaron fueron los siguientes: un chat para permitir la comunicación de modo síncrono; un foro para permitir la comunicación de modo asíncrono; un *pool* de votaciones que permita sondear, cuestionar, consensuar, etc., a los estudiantes. Además, se hace uso de la programación por pares para la resolución de problemas de programación de forma colaborativa. En los resultados obtenidos, los estudiantes consideran al chat como la herramienta colaborativa que les permite comunicarse con el compañero de forma más inmediata, permitiendo un diálogo más ágil y menos estructurado. Sin embargo, aprender a programar en un entorno como Eclipse, bajo un lenguaje de propósito general como Java puede ser más complicado para los novatos, ya que en lugar de hacer hincapié en los principios fundamentales de la programación, los lenguajes de propósito general se pierden en nociones secundarias y sutilezas concernientes a la implementación. El trabajo de Jurado al igual que el de nosotros está relacionado con la enseñanza de programación en jóvenes, pero a diferencia de nuestro trabajo está diseñado para estudiantes con conocimientos previos y no para novatos.

Chen-Chung et al. [16] En este artículo, los autores proponen un enfoque, mediante el cual se utiliza un juego de simulación para ayudar a los estudiantes de primer semestre de universidad a desarrollar sus habilidades para resolver problemas computacionales. Estos juegos de simulación abordan las características de la solución de problemas a través de la construcción y la simulación de la experiencia. El juego de simulación de este estudio ayuda a los estudiantes a aprender con la simulación. Se encontró que los estudiantes, al aprender a resolver problemas computacionales con juegos, son más propensos a percibir una mejor experiencia de aprendizaje que en las clases tradicionales. Los resultados mostrados en este estudio demuestran que los juegos de simulación constituyen un método eficaz para ayudar a los programadores novatos para aprender habilidades de resolución de problemas.

En este trabajo se realiza un juego de simulación lúdico para ayudar a los programadores novatos (de primer semestre de universidad) a aprender habilidades de resolución de problemas con un método eficaz pero no emplea la colaboración en sus estudios de investigación y práctica.

Wolz et al. [23] Los autores utilizaron el entorno Scratch para crear todo tipo de entornos 2D que involucraran procesos asincrónicos y concurrentes, como videojuegos, historietas, tarjetas animadas, parodias de televisión, tutoriales educativos, simulaciones científicas, entre otros; la investigación se realizó con estudiantes que iniciaban el curso de Ciencias de la computación en la universidad. Los autores presentaron una justificación de integración del trabajo con Scratch en el currículo de Ciencias de la computación para los primeros

grados de universidad, basados en la capacidad de generar una transición más amigable en la enseñanza de la programación para lenguajes más robustos como Java o C. El trabajo de Wolz al igual que el de nosotros está relacionado con la enseñanza de la programación en jóvenes universitarios, pero a diferencia del trabajo nuestro no hacen uso de la colaboración ni de los elementos de la lúdica o juegos para mejorar el proceso de aprendizaje.

Rizvi et al. [24] utilizaron el entorno Scratch para crear animaciones, proyectos multimedia y juegos con el fin de aplicarlo en el curso inicial de Ciencias de la computación y determinar su eficacia en la mejora de la retención, el rendimiento y la actitud en cuanto a habilidades de programación. En esta experiencia participaron alumnos que recién se matricularon a la universidad y tomaron el curso referido. En particular, centraron su atención en estudiantes “en riesgo”, es decir, con una preparación muy débil en matemáticas. Con ellos se buscó trabajar en la elaboración de varios proyectos multimedia, los cuales involucraron el uso de una alta cantidad de gráficos, elementos animados y sonidos. Nuestro trabajo sigue esta misma línea de investigación, pero intenta evaluar y aplicar la utilización de una metodología colaborativa.

En la tabla 3 se resume las características más relevantes de estudios relacionados, identificando claramente cada uno de los aspectos que aportan al trabajo actual:

Propuesta	Modelo /Plataforma	Población Objetivo	Colaboración			Lúdica		
			PI	IA	E P	D	M	C
Modelo “ChildProgramming”[1]	Scratch	Niños de edades entre 8-11 años	X			X		
“Learning Computer Science Concepts with Scratch”[13]	Scratch	Jóvenes de edades entre 14-15 años						X
“Digispired: Digital inspiration for interactive game design and programming” [14]	LEGO	Niños de edades entre 11-13						X

Propuesta	Modelo /Plataforma	Población Objetivo	Colaboración			Lúdica		
			PI	IA	EP	D	M	C
“Cole-Programming: Incorporando Soporte al Aprendizaje Colaborativo en Eclipse” [15]	Eclipse	Estudiantes de primer semestre de universidad	X				X	
“The effect of simulation games on the learning of computational problem solving”[16]	Train B&P	Estudiantes de primer semestre de universidad						X
“Starting with scratch in cs1” [23]	Scratch	Estudiantes de primer semestre de universidad						X
“A cs0 course using Scratch”[24]	Scratch	Estudiantes de primer semestre de universidad						X

Tabla 3. Comparación de Modelos

En la **Tabla 3** se muestra una taxonomía en la cual se hace referencia a cuatro criterios de comparación de los diferentes trabajos encontrados en el estado del arte:

- **Modelo/Plataforma:** Hace referencia a los modelos propuestos por parte del investigador (es) o las plataformas usadas para la puesta en práctica de sus investigaciones.
- **Población Objetivo:** Hace referencia a la población a la cual va dirigida la investigación y en la cual se realizaron las pruebas.
- **Colaboración:** Hace referencia al uso o no uso de la colaboración en su implementación mediante la verificación de los tres principios colaborativos básicos: Interdependencia positiva (PI), Responsabilidad individual (IA), Igual Participación (EP).

- **Lúdica⁸**: Hace referencia al uso o no uso de la lúdica en su implementación mediante la verificación de las características de los juegos: Dinámicas (D), Mecánicas (M), Componentes (C).

En la anterior taxonomía (**Tabla 3**) se puede observar que se han planteado diferentes modelos a poblaciones, pero a grandes rasgos no se ha propuesto aún un modelo colaborativo y lúdico dirigido directamente a jóvenes; por lo tanto en este proyecto se planteará un modelo que abarque los anteriores aspectos dirigido a adolescentes de 14 a 19 años, así como también definir las características que componen el mecanismo de enseñanza. De esta manera se logra incorporar los aspectos colaborativo y lúdico en la experiencia, tanto de la enseñanza como el aprendizaje de la programación como se muestra en la **Tabla 4**.

Propuesta	Modelo /Plataforma	Población Objetivo	Colaboración			Lúdica		
			PI	IA	E P	D	M	C
“Young Programming”	Scratch	Jóvenes de 14 a 19 años	X	X	X	X	X	X

Tabla 4. Modelo YoungProgramming.

⁸ Lúdica: Acción que produce diversión, placer y alegría.

Capítulo 3

En este capítulo se describirá el método de aplicación de la lúdica y la colaboración para el proceso de enseñanza y aprendizaje de la programación (ver **Figura 5**). En la primera parte se abarcarán las técnicas de aprendizaje colaborativo, mediante las cuales se llegará a la creación de **LudiCop**, una técnica lúdica y colaborativa para la enseñanza de la programación, detallada en la sección 3.1. Después se procederá al modelado colaborativo formal de la técnica propuesta mediante CIAM, realizado en la sección 3.2.

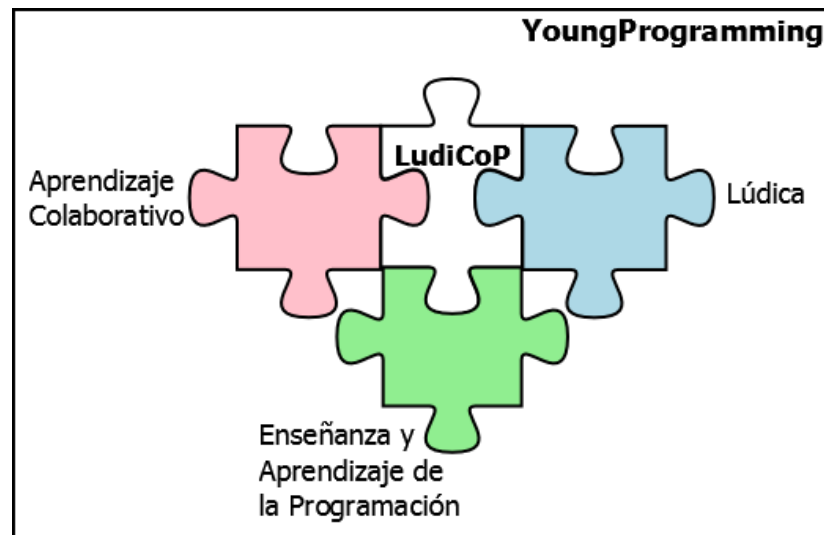


Figura 5. Contextualización del capítulo 3

3.1 MÉTODO DE APLICACIÓN DE LA LÚDICA Y LA COLABORACIÓN

Las técnicas de AC en las que se focaliza este trabajo son aquellas aplicables a grupos formales [34]. Para las técnicas que cumplen con esta condición, el esquema general que se debería seguir para su aplicación se muestra en la **Figura 6**.

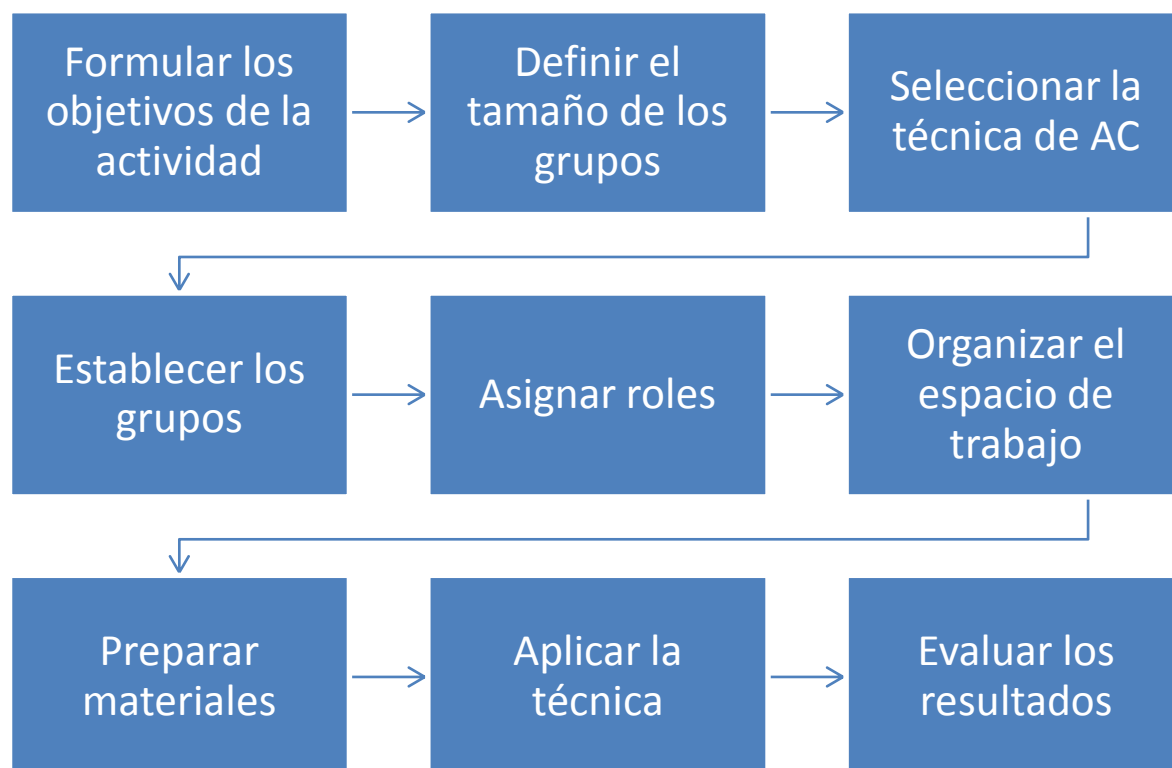


Figura 6. Esquema general de la aplicación de AC. Fuente: [50]

Formular objetivos de la actividad

Se debe plantear claramente cuáles son los propósitos pedagógicos y de investigación perseguidos por la actividad. La definición clara y el entendimiento de los objetivos perseguidos, previos al inicio de la actividad, contribuyen fuertemente al éxito de la misma. Los objetivos se definen de manera que los estudiantes deban percibir que pueden lograr su aprendizaje si y solo si los otros integrantes del grupo alcanzan sus objetivos [51].

Definir el tamaño de los grupos

Johnson y Johnson [34] plantean el uso de grupos pequeños, de al menos dos personas para poder aplicar el aprendizaje colaborativo. Dillenbourg [48] plantea que algunos autores indican la efectividad de aprendizaje colaborativo se realiza en grupos pequeños, que contienen entre dos a cinco miembros. Sin embargo, cada una de las técnicas tiene su propio valor mínimo y máximo (Dentro del rango). Igualmente, en última instancia es potestad del docente definir este valor según sea el número de estudiantes, la mecánica de trabajo y el tiempo que dispone para realizar la actividad, formando preferiblemente grupos seleccionados por él y que sean heterogéneos permitiendo que los alumnos tengan acceso a diversas perspectivas y métodos de resolución de problemas, y produciendo un mayor desequilibrio cognitivo, necesario para estimular el aprendizaje y desarrollar la responsabilidad individual [51].

Seleccionar la técnica de AC

El tercer paso es en el que se centra esta tesis y consiste en la selección de la técnica de AC que será utilizada para llevar a cabo el proceso de enseñanza de la programación. Para esto se parte de un conjunto de técnicas las cuales fueron elegidas luego de una extensa revisión de la literatura velando porque se cumplieran los siguientes criterios:

- Las técnicas deben tener una estructura bien descrita, que permita realizar en la caracterización el conjunto de métodos de instrucción de las actividades que deben hacer tanto los estudiantes como los docentes [50].
- Las técnicas deben explicar claramente el rol del docente y el del estudiante, lo cual permite evitar los temores al cambio [38].

Un listado de las técnicas consideradas se presenta en la tabla 5. El procedimiento de selección como tal se basa en los elementos definidos en los dos pasos anteriores, además se verifica si la técnica encontrada cumple con alguno de los principios de colaboración [38] y si además contienen un componente lúdico: Interdependencia positiva(PI), Responsabilidad individual (IA) ,Iguar Participación (EP) y Lúdica (L).

Técnica de AC	PI	IA	EP	L
<p>Jigsaw (Rompecabezas): Los estudiantes trabajan en pequeños grupos con objeto de desarrollar sus conocimientos sobre un tema determinado y formular métodos eficaces de transmitírselos a otros. A continuación, estos grupos de “expertos” se deshacen y los estudiantes pasan a nuevos grupos “rompecabezas”, formado cada uno de ellos por alumnos que han llegado a dominar distintos subtemas. Esta técnica es útil para motivar a los estudiantes a que asuman la responsabilidad de aprender algo suficientemente bien como para enseñárselo a sus compañeros. Ofrece también la oportunidad a cada alumno de constituirse en el centro de atención. Cuando los estudiantes asumen el rol de profesor, dirigen el diálogo, de manera que incluso los más reticentes a hablar en público deben asumir roles de líderes [53].</p>	X		X	
<p>Numbered Heads Together (Cabezas Numeradas Simultaneas): Mediante esta técnica todos los participantes necesitan saber y estar listos para explicar las respuestas del grupo. Como los estudiantes ayudan a sus compañeros de clase, se ayudan a sí mismos y al grupo, porque la respuesta dada pertenece a todos, no solamente al que la está elaborando. En esta técnica, cada estudiante de cada grupo se enumera (1 a 4), luego el profesor o un estudiante hace una pregunta acerca del tema en estudio. Los grupos se juntan para dar una respuesta y finalmente el profesor selecciona un número entre 1 y 4 y la persona que lo tenga explica la respuesta [52].</p>	X			

Técnica de AC	PI	IA	EP	L
<p>Pairs-Check (Chequeo de Pares): Un integrante de cada grupo desarrolla su solución para un problema, pensando en voz alta. Los otros integrantes del grupo escuchan, observan, y proveen retroalimentación a las soluciones de los demás. El observador facilita a la persona encargada de la pregunta. Se continua con la siguiente pregunta, rotando los roles, hasta terminar todas las preguntas.[52]</p>	X	X		
<p>Think-Pair-Square (Piensa-Pares-Comparte): Consiste en una estructura cooperativa de tres pasos. Durante el primer paso, las personas piensan en silencio acerca de una pregunta formulada por el instructor. Los individuos se emparejan durante el segundo paso e intercambian sus pensamientos con los otros. En el tercer paso, los pares comparten sus respuestas con otros pares, otros equipos, o todo el grupo [52].</p>	X			
<p>Team-Pair-Solo (Equipo-Pares-Solo): Los estudiantes hacen problemas por primera vez como un equipo, a continuación, con un compañero, y, finalmente, por su propia cuenta. Está diseñado para motivar a los estudiantes para hacer frente a los problemas y tener éxito en lo que inicialmente estaba más allá de sus capacidades. Se basa en una simple noción de aprendizaje mediado. Los estudiantes pueden hacer más cosas con ayuda (mediación) de lo que pueden hacer por sí solos. Por lo que se les permite trabajar en problemas que no podían hacer por sí solos, por primera vez como un equipo y luego con un compañero, progresando hasta el punto de hacer por si solos lo que en un principio podían hacer sólo con la ayuda de sus compañeros [52].</p>		X		
<p>Para hablar, paga ficha: Los estudiantes participan en un dialogo en grupo, entregando una ficha cada vez que intervienen. La finalidad de esta técnica es asegurar una actuación equitativa regulando la frecuencia con la que se permite hablar a cada miembro del grupo. Como prioriza la participación plena y equitativa de todos los miembros, esta técnica estimula a los estudiantes reticentes a hablar y a los habladores a reflexionar. Esta técnica es útil para ayudar a los alumnos a comentar cuestiones discutidas y también para resolver problemas de comunicación o de proceso, como los relativos al predominio en el grupo o a los choques entre los miembros del mismo [53].</p>			X	

Técnica de AC	PI	IA	EP	L
<p>Resolución de Problemas por parejas pensando en voz alta (RPPPVA): Las parejas de estudiantes reciben una serie de problemas así como unos roles específicos –solucionador de problemas y oyente– que se intercambian en cada problema. El solucionador de problemas “piensa en voz alta” y habla mientras va dando los pasos necesarios para resolverlo. El compañero que escucha, sigue los pasos, trata de comprender el razonamiento que subyace a los pasos y hace sugerencias si hay errores.</p> <p>Articular los propios procesos de resolución de problemas y escuchar con atención los pasos del otro ayuda a los estudiantes a practicar lo que han leído u oído en una lección magistral. Esta técnica mejora las competencias analíticas ayudando a los estudiantes a formular ideas, probar los conceptos, comprender la secuencia de pasos subyacente a su pensamiento y a identificar errores en el pensamiento de otros [53].</p>	X	X		
<p>Resolución estructurada de problemas: Proporciona a los estudiantes un procedimiento para resolver un problema complejo, relacionado con los contenidos, en un periodo de tiempo especificado de antemano. Todos los miembros del grupo tienen que ponerse de acuerdo en la solución y deben ser capaces de explicar tanto la respuesta como la estrategia utilizada para resolver el problema. Esta técnica ayuda a los alumnos a dividir el proceso en pasos concretos. Así, los estudiantes aprenden a identificar, analizar y resolver problemas de un modo organizado. En vez de sentirse abrumados por la magnitud de un problema, esta actividad da a los alumnos una pauta a la que atenerse, de manera que sepan por dónde empezar. Al facilitarles una serie de pasos manejables, evita que los estudiantes se pierdan o emprendan tareas irrelevantes [53].</p>	X			

Tabla 5. Comparación de Técnicas de AC

Las técnicas descritas anteriormente no cumplen con todos los criterios propuestos, por lo tanto se propone una nueva técnica de aprendizaje colaborativo para llevar a cabo el proceso de enseñanza de la programación, considerando aspectos lúdicos y la edad de la población objetivo, en este caso jóvenes. La técnica propuesta es la siguiente:

LudiCoP: Esta técnica es una propuesta que parte de la unión de tres técnicas de aprendizaje colaborativo como lo son Jigsaw, RoundRobin y RPPPVA [53], con la incorporación de aspectos lúdicos para llevar a cabo el proceso de enseñanza de la programación en jóvenes. En la parte inicial se utiliza la técnica Jigsaw con el objeto de que los alumnos desarrollen sus conocimientos sobre un tema determinado (solucionando un problema) y que sean capaces de formular métodos eficaces de transmitírselos a otros. A continuación, estos grupos de “expertos” se deshacen y los estudiantes pasan a nuevos grupos “rompecabezas”, formado cada uno de ellos por alumnos que han llegado a dominar

distintos subtemas. Después, se formula un problema de tal forma que cada uno de los integrantes del grupo interaccionen (uniendo sus partes) para lograr una solución pero esto lo hacen de forma que cada integrante del grupo programe (en voz alta) durante un intervalo de tiempo, compartiendo así su conocimiento con los demás, y rotando hasta que todos los integrantes del grupo programen (RoundRobin). Este proceso de rotación se acaba cuando se logre solucionar el problema.

En la sección 3.1 se realiza una descripción detallada de esta técnica.

Establecer los grupos

Aquí el docente debe formar preferiblemente grupos seleccionados por él, tratando de que se obtengan grupos con los mismos niveles de conocimiento. Permitiendo así que los alumnos tengan acceso a diversas perspectivas y métodos de resolución de problemas, y produciendo un mayor desequilibrio cognitivo, necesario para estimular el aprendizaje y desarrollar la responsabilidad individual [51].

Es necesario identificar al grupo con un nombre, logotipo y slogan. Ya que la identidad es un factor que apoya la consecución de objetivos en grupo. La identificación por parte de los integrantes del equipo se hace con el fin de alcanzar la (interdependencia de identidad) [51].

Asignar roles

Asignar roles a los alumnos es una de las maneras más eficaces de asegurarse de que cada integrante del grupo tenga asignada una responsabilidad (Responsabilidad individual) y de que los miembros del grupo trabajen juntos sin tropiezos y en forma productiva. Crea interdependencia entre los miembros del grupo cuando se les asignan roles complementarios interconectados (Interdependencia positiva). Aquí el docente ejecuta las decisiones acerca de los roles que desarrollaran la actividad [51].

Organizar el espacio de trabajo

Este paso aplica tanto para actividades presenciales como virtuales o mixtas, e implica que el docente vele para que cada grupo tenga un espacio adecuado para la interacción de los estudiantes según los roles requeridos. En el caso de las actividades virtuales y mixtas, este paso incluye la preparación de las herramientas computacionales que serán utilizadas de entre las que son aplicables para la técnica [50].

Preparar los materiales

Cuando los grupos ya estén formados y cada estudiante tiene asignado un rol, es tiempo de asignar los materiales que permiten que los grupos alcancen los objetivos y metas propuestas para la actividad (interdependencia de materiales). Esta actividad Indica la forma en que serán entregados los recursos a cada grupo una vez estén formados, con el propósito de crear aspectos que favorezcan la colaboración entre los alumnos (Responsabilidad individual) [51].

Aplicar la técnica

Una vez se han cumplido los pasos anteriores, el profesor puede proceder a aplicar la técnica. Durante el desarrollo de la misma debe además hacer un monitoreo constante y, en caso en que la técnica lo disponga, intervenir según el rol que ejerza en la misma [50].

Evaluar los resultados

Como paso final, luego de finalizada la técnica, el profesor debería realizar una evaluación de los resultados obtenidos por cada grupo. Tal evaluación puede ser cualitativa o cuantitativa y debería brindar a los estudiantes una realimentación sobre la obtención de los objetivos planteados [50].

3.2 LudiCoP (Ludic-Collaborative Programming). Aspectos Colaborativos

Esta técnica es una propuesta que parte de la unión de tres técnicas de aprendizaje colaborativo como lo son Jigsaw, RoundRobin y RPPPVA [53], con la incorporación de aspectos lúdicos para llevar a cabo el proceso de enseñanza de la programación en jóvenes. En la Tabla 6 se presenta una descripción detallada de todo el procedimiento para llevar a cabo la técnica propuesta, basándose en [51].

Procedimiento					
Actividad	Descripción	PI	IA	EP	L
Describir brevemente la actividad de aprendizaje	La idea en si es que el docente explique y describa la actividad en términos de objetivos, criterios de éxito, tareas, duración, roles, evaluación, y expectativas frente a los estudiantes (Responsabilidad Individual).		X		
Especificar reglas de la actividad	Representa las reglas y restricciones que permiten al docente realizar un control sobre comportamientos, fechas y circunstancias de la actividad, un ejemplo de ello puede ser garantizar la igual participación de todos los estudiantes (Igual Participación).			X	
Definir especificación de recompensas comunes para el grupo	Como una forma de generar (Interdependencia Positiva) en el grupo de estudiantes, se pueden definir recompensas que serán otorgadas siempre y cuando el grupo coopere para alcanzar los objetivos comunes de la clase.	X			
Formar grupos de 3 estudiantes.	Aquí los estudiantes se agrupan según el criterio del docente, buscando que el profesor sea quien escoja cada uno de los integrantes con el fin de obtener grupos homogéneos a			X	

Procedimiento					
Actividad	Descripción	PI	IA	EP	L
	nivel de conocimientos. Esto con el fin de promover la (Igual Participación).				
Identificar al grupo con un nombre, logotipo, slogan	La identidad es un factor que apoya la consecución de objetivos en grupo. Esta actividad pretende que los integrantes del equipo de trabajo, propongan un nombre, logotipo y slogan que los identifique para alcanzar así la (interdependencia de identidad).	X	X	X	
Asignar roles	Asignar roles a los alumnos es una de las maneras más eficaces de asegurarse de que cada integrante del grupo tenga asignada una responsabilidad (Responsabilidad individual) y de que los miembros del grupo trabajen juntos sin tropiezos y en forma productiva. Crea interdependencia entre los miembros del grupo cuando se les asignan roles complementarios interconectados (Interdependencia positiva). Aquí el docente ejecuta las decisiones acerca de los roles que desarrollaran la actividad.	X	X		
Exponer las unidades temáticas	El docente imparte, al grupo general, el conocimiento de las nociones básicas concernientes a la programación.				X
Formar grupos expertos (JigSaw)	Cada uno de los integrantes de los grupos originales pasará a ser un nuevo integrante de un nuevo grupo “experto” con el objeto de que los alumnos desarrollen sus conocimientos sobre un tema determinado (solucionando un problema) y que sean capaces de formular métodos eficaces de transmitírselos a otros. En ese proceso de crear mecanismos para retransmitir sus conocimientos adquiridos a sus compañeros, el estudiante terminará de afianzar su conocimiento como experto en la temática determinada.	X	X		

Procedimiento					
Actividad	Descripción	PI	IA	EP	L
Retornar a sus grupos originales	Los grupos de “expertos” se deshacen y los estudiantes pasan de nuevo a sus grupos originales “rompecabezas”, formado cada uno de ellos por alumnos que han llegado a dominar distintos subtemas.				
Programar en voz alta (RPPPV & RoundRobin)	Cada integrante del grupo original programa (en voz alta), de acuerdo al conocimiento adquirido como “experto”, durante un intervalo de tiempo, compartiendo así su conocimiento con los demás (solucionando un problema). Rotando hasta que todos los integrantes del grupo programen (RoundRobin). Este proceso de rotación se acaba cuando se logre solucionar el problema propuesto.	X	X	X	
Mantener el momento de colaboración	Esta actividad reúne estrategias para controlar, supervisar e intervenir en caso de problemas en el grupo y de esta forma facilitar al docente la labor de mantener la colaboración en el grupo, con colaboración se entienden criterios como: (Responsabilidad individual, Igual participación e Interdependencia positiva).	X	X	X	
Realizar una evaluación sumativa	Este tipo de evaluación permite al docente testear el nivel final de conocimientos, a nivel grupal e individual, y de esta forma medir el nivel de (Responsabilidad Individual) de cada estudiante.		X		

Procedimiento					
Actividad	Descripción	PI	IA	EP	L
Hacer que los grupos comparen entre sí sus resultados	Para dar cierre a la actividad, es importante que los estudiantes puedan organizar su conocimiento, formularlo y explicárselo a los demás, esto según Johnson y Johnson [4], esta es una forma más de reafirmar el conocimiento adquirido y de generar (Interdependencia positiva), ya que los alumnos requieren de sus compañeros para completar su conocimiento.	X			

Tabla 6. Procedimiento LudiCoP

3.3 LudiCoP (Ludic-Collaborative Programming). Aspectos Lúdicos.

La sección 2.7.2 del presente trabajo hace alusión a los elementos fundamentales dentro de las actividades lúdicas. Estos elementos son: Las dinámicas, las mecánicas y los componentes del juego.

Las dinámicas y componentes se van a describir a continuación, mientras que las mecánicas de juego serán descritas en el capítulo 4, ya que éstas se asocian al proceso YoungProgramming como parte de la dimensión lúdica.

Dinámicas de Juego de LudiCoP: Aquí se definen las restricciones o reglas (véase **Tabla 7**) bajo las cuales se llevará a cabo la actividad. Se describen tanto los tiempos de cada sub-actividad como el tiempo de duración de cada ronda (RoundRobin). También se define el tamaño de los grupos y la responsabilidad de cada integrante del equipo.

Componentes de Juego de LudiCoP: Se establecen recompensas o puntos (véanse **Tabla 8 y Tabla 9**) que pueden ser medidas en función al esfuerzo o a las tareas completadas. Además, se contará con una tabla de clasificación, en donde se verán reflejados los puntos acumulados por cada grupo.

Las dinámicas y los componentes que serán incluidos en este trabajo son los siguientes:

Nombre de la Dinámica: Reglas
<p>Objetivo en LudiCoP:</p> <p>Es importante definir las reglas del juego para realizar un control sobre los comportamientos y circunstancias dentro de la actividad.</p>

Tabla 7. Dinámica “Reglas” dentro de LudiCoP. Tomado de [69].

Nombre del Componente: Competencia
Objetivo en LudiCoP: La competencia genera aumento en el entusiasmo por terminar las tareas antes que otros, por ganar puntos, otros reconocimientos para quedar en primeros lugares.

Tabla 8. Dinámica “Competencia” dentro de LudiCoP Tomado de [69].

Nombre del Componente: Recompensa	
Objetivo en LudiCoP: Es importante para mantener motivados y comprometidos a los estudiantes. Dependiendo del comportamiento o del desempeño se brindan las recompensas.	
Utilización en LudiCoP:	Las recompensas dentro de LudiCoP son certificados de participación de la actividad, además, el grupo ganador podrá programar un robot “Mbot” gracias al conocimiento adquirido durante toda la experiencia de aprendizaje.

Tabla 9. Componente “Recompensa” dentro de LudiCoP. Tomado de [69].

3.4 Modelamiento de LudiCoP a través de CIAM

CIAM es una metodología que simplifica el proceso de colaboración en tres fases principales: grupos, procesos y usuarios. Al ser una metodología que se enfoca en los miembros y las responsabilidades es la mejor opción para modelar de manera formal la técnica propuesta. Cabe resaltar que como no se presenta ningún tipo de interacción con una aplicación, los objetos del modelo del dominio presentados en CIAM [54], serán sustituidos por objetos creados en la interacción entre los integrantes que forman parte de la organización. Por lo tanto, tampoco será abarcada la fase 5(modelado de interacción) dentro de CIAM.

1. Creación del Sociograma.

En esta fase se modela la estructura de la organización, así como las relaciones que existen entre los distintos integrantes de la misma. Los integrantes que forman la organización podrán entrar en una de las siguientes categorías: roles, actores, agentes software; o agrupaciones de los anteriores, dando lugar a grupos o equipos de trabajo.

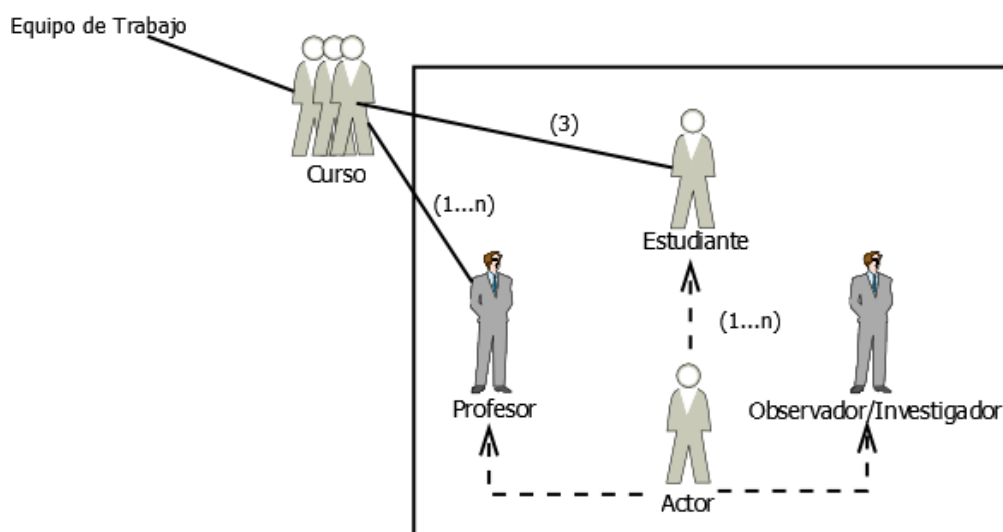


Figura 7. Sociograma de LudiCoP. Elaboración propia, basándose en [54].

En este entorno, detallado en la **Figura 7** se contará con tres actores principales (Profesor, Estudiante, Observador/Investigador) para que se lleven a cabo las tareas o actividades propuestas por la técnica. Se muestran las relaciones necesarias para llevar a cabo la colaboración, cooperación y comunicación entre cada uno de los actores, teniendo en cuenta el desempeño de cada una de las tareas por parte de los actores.

2. Creación del modelado de Responsabilidades.

En esta fase se presta atención a la perspectiva individual de cada uno de sus integrantes (roles) de la organización, añadiendo a sus responsabilidades compartidas las que le sean exclusivas. Se realiza primero la tabla de partición y posteriormente se realiza el modelado de responsabilidades por cada uno de los roles establecidos en el Sociograma.

Tareas/Roles	Profesor	Estudiante	Observador	Tipo
Describir brevemente la actividad de aprendizaje	X			
Especificar reglas de la actividad	X			
Definir especificación de recompensas comunes para el grupo	X			
Formar grupos de 3 estudiantes	X	X		
Identificar al grupo con un nombre, logotipo, slogan		X		
Asignar roles	X	X		












Tareas/Roles	Profesor	Estudiante	Observador	Tipo
Exponer las unidades temáticas	X	X		
Formar grupos expertos(JigSaw)		X	X	
Retornar a sus grupos originales		X		
Programar en voz alta (RPPPVA & RoundRobin)		X	X	
Mantener el momento de colaboración	X	X	X	
Realizar una evaluación sumativa	X	X	X	 
Hacer que los grupos comparen entre sí sus resultados	X	X		

Tabla 10. Tabla de partición CIAM elaborada a partir de Tabla 6

La información expresada mediante las técnicas anteriores sirve de base para la definición del modelo de responsabilidades asociado a cada uno de los roles del sistema (Profesor, Estudiante, Observador/Investigador). Tal como se muestra en las tablas 11,12 y 13.

Nombre de la Responsabilidad	Tipo de Tarea	Objeto del modelo del dominio	Pre-requisitos	
			Tarea	Datos
Describir brevemente la actividad de aprendizaje		C/L: Contenido	INI	
Especificar reglas de la actividad		C/L/E:Reglas	Descripción de la actividad de aprendizaje	
Definir especificación de recompensas comunes para el grupo		C/L/E: Recompensas	-Descripción de la actividad de aprendizaje -Reglas de la Actividad	-Contenido -Reglas
				-Contenido










Nombre de la Responsabilidad	Tipo de Tarea	Objeto del modelo del dominio	Pre-requisitos	
			Tarea	Datos
Formar grupos de 3 estudiantes		C/L/E: Criterios de Selección	-Descripción de la actividad de aprendizaje -Reglas de la Actividad	-Reglas
Asignar roles		C/L/E: Responsabilidades	-Formar grupos de 3 estudiantes	-Criterios de Selección
Exponer las unidades temáticas		C/L/E:Contenido		-Contenido
Mantener el momento de colaboración		C/L/E: Estrategias	Cualquier actividad	
Realizar una evaluación sumativa	 	C/L/E:Evaluación	-Formar grupos expertos -Retornar a sus grupos originales -Programar en voz alta	-Contenido -Reglas
Hacer que los grupos comparen entre sí sus resultados		C/L/E: Instrucciones	-Realizar una evaluación sumativa	-Contenido

Tabla 11. Modelo de Responsabilidades del Rol *Profesor*

Ahora se procede a realizar el modelado de Responsabilidades del Rol Estudiante.

Nombre de la Responsabilidad	Tipo de Tarea	Objeto del modelo del dominio	Pre-requisitos	
			Tarea	Datos
Formar grupos de 3 estudiantes		L:Criterios de Selección	-Descripción de la actividad de aprendizaje -Reglas de la Actividad	-Contenido -Reglas
Identificar al grupo con un nombre, logotipo, slogan		C/L/E:Grupo	-Formar grupos de 3 estudiantes	










Nombre de la Responsabilidad	Tipo de Tarea	Objeto del modelo del dominio	Pre-requisitos	
			Tarea	Datos
Asignar roles		L:Responsabilidades	-Formar grupos de 3 estudiantes -Identificar al grupo	-Grupo
Exponer las unidades temáticas		L:Contenido		-Contenido
Formar grupos expertos(JigSaw)		L:Material C/L/E: Solución	-Formar grupos de 3 estudiantes -Identificar al grupo	-Contenido -Reglas
Retornar a sus grupos originales			-Formar grupos expertos	-Grupo
Programar en voz alta (RPPPVA & RoundRobin)		L:Problema C/E:Solución	-Formar grupos expertos -Retornar a sus grupos originales	-Contenido -Reglas -Grupo -Problema
Mantener el momento de colaboración		L:Estrategias	Cualquier actividad	-Estrategias
Realizar una evaluación sumativa	 	L:Evaluación	-Formar grupos expertos -Retornar a sus grupos originales -Programar en voz alta	-Contenido -Reglas -Evaluación
Hacer que los grupos comparen entre sí sus resultados		L:Solución L:Instrucciones	-Realizar una evaluación sumativa	-Grupo -Problema -Solución

Tabla 12. Modelo de Responsabilidades del Rol *Estudiante*

Y por último se modelan las Responsabilidades del rol Observador/Investigador.





Nombre de la Responsabilidad	Tipo de Tarea	Objeto del modelo del dominio	Pre-requisitos	
			Tarea	Datos
Formar grupos expertos(JigSaw)		C/L/E: Observaciones	-Formar grupos de 3 estudiantes al grupo -Identificar al grupo	-Contenido -Reglas -Grupo
Programar en voz alta (RPPPVA & RoundRobin)		C/L/E: Observaciones	-Formar grupos de 3 estudiantes al grupo -Identificar al grupo	-Contenido -Reglas -Grupo
Mantener el momento de colaboración		C/L/E: Pautas	Cualquier actividad	
Realizar una evaluación sumativa		C/L/E: Observaciones	-Formar grupos de 3 estudiantes al grupo -Identificar al grupo -Programar en voz alta	-Contenido -Reglas -Grupo

Tabla 13. Modelo de Responsabilidades del Rol *Observador/Investigador*

3. Creación del Modelo de Inter-Acción.

Este modelo permite especificar el funcionamiento completo del proceso de grupo que puede ser cooperativo, colaborativo o mixto. Este modelo se define mediante un diagrama de estados que permitirá relacionar toda la información definida mediante las dos técnicas anteriores. Este diagrama se representa mediante un grafo cuyos nodos son las actividades que componen el trabajo en grupo y cuyos arcos indican relaciones entre dichas actividades (de orden, de dependencia de datos, de condición, notificación, de paso del tiempo, etc.). Cabe resaltar que como no se está interactuando con ningún tipo de aplicación, los objetos de Creación, Lectura y Escritura (C/L/E) se reemplazan por objetos de tipo conceptual como se puede observar en la **Figura 8**.

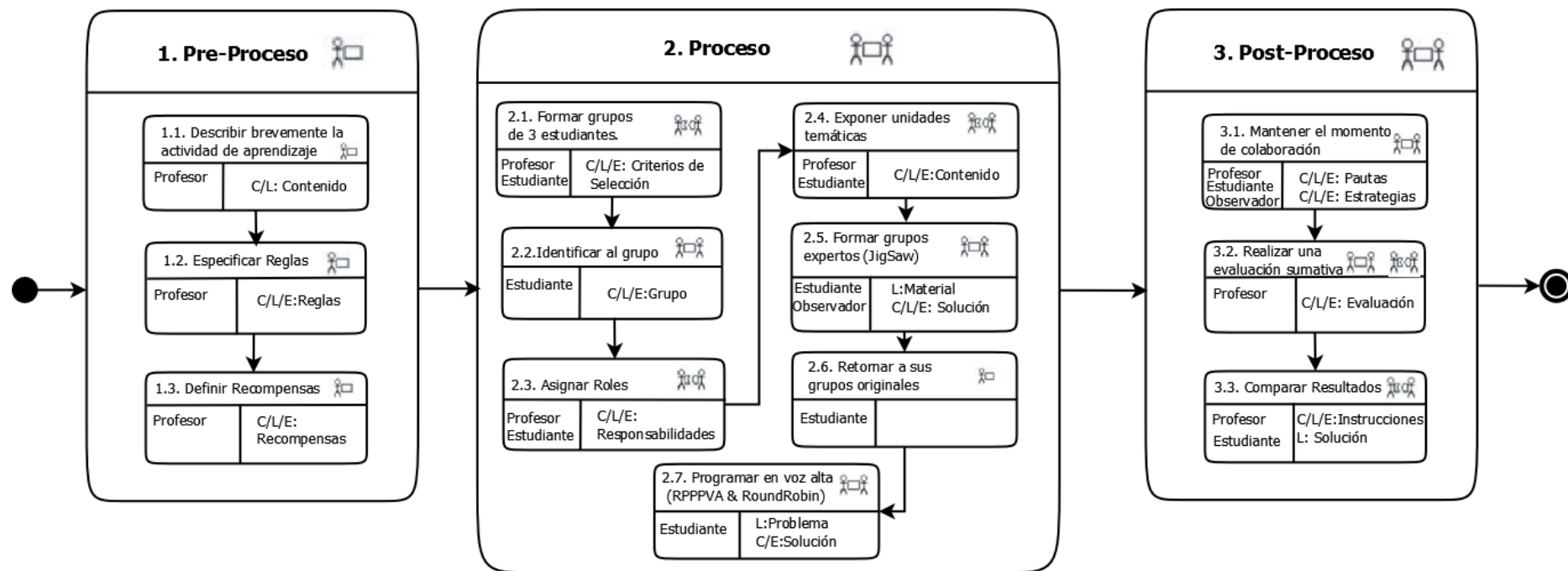


Figura 8. Modelo de Inter-Acción de LudiCoP. Elaboración Propia.

4. Modelado de Tareas de Trabajo en Grupo.

En esta fase se aumenta el nivel de detalle con el que se especifican las tareas de grupo (colaborativas o cooperativas) identificadas previamente. Es importante destacar la necesidad de modelar de forma diferenciada las tareas cooperativas de las colaborativas. La información relevante en cada una de ellas varía. Teniendo en cuenta la definición de Dillenbourg [3] esta distinción se traduce en dos aspectos importantes: la división de tareas (en tareas individuales en el caso de la cooperación) y en los objetos manejados (comunes en el caso de la colaboración). Cabe resaltar que para este proyecto, los actores no interaccionarán con ningún tipo de aplicación, por lo tanto, en el modelado de tareas cooperativas y colaborativas solo se va a comprender la división de tareas y no los objetos manejados, es decir, se va a realizar división de tareas tanto para las tareas de tipo cooperativo como para las de naturaleza colaborativa.

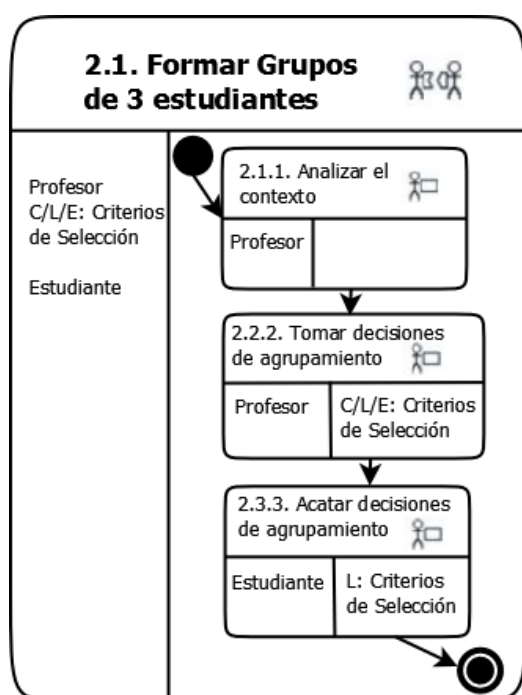


Figura 9. Tarea cooperativa 2.1

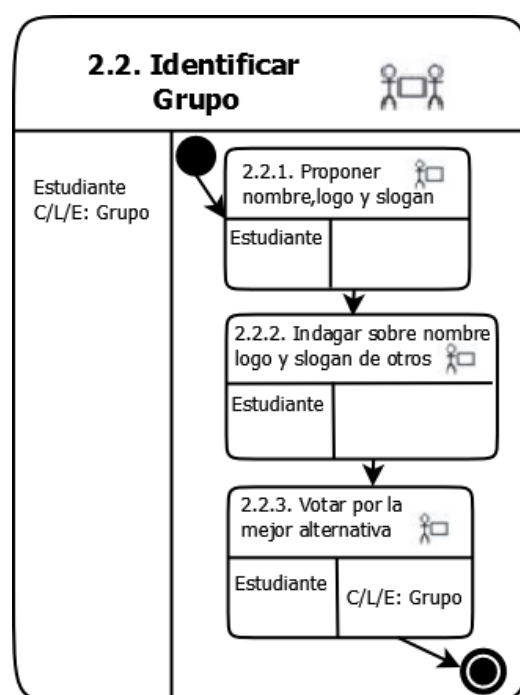


Figura 10. Tarea colaborativa 2.2

En la actividad 2.1.1 (véase **Figura 9**) el Profesor realiza un análisis del contexto para posteriormente tomar decisiones de agrupamiento, donde preferiblemente debe formar grupos seleccionados por él y que sean heterogéneos permitiendo que los estudiantes tengan acceso a diversas perspectivas y métodos de resolución de problemas, y produciendo un mayor desequilibrio cognitivo, necesario para estimular el aprendizaje y desarrollar la responsabilidad individual [51]. Después de esto, los estudiantes según los criterios de selección propuestos por el Profesor, proceden a formar los grupos de trabajo pertinentes.

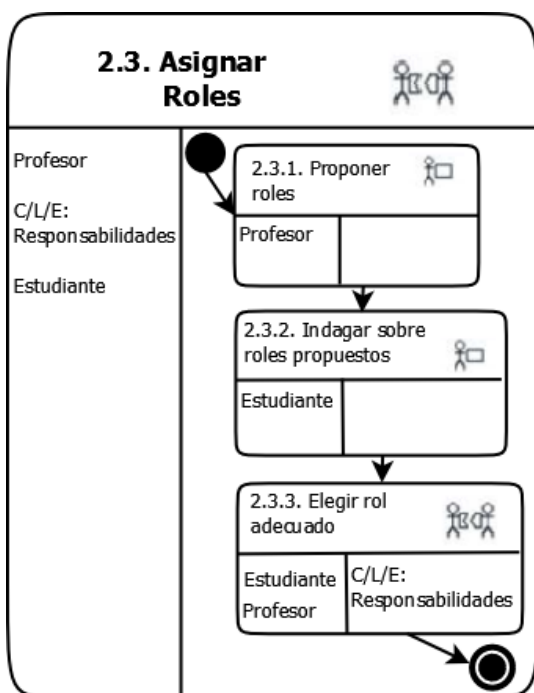


Figura 11. Tarea cooperativa 2.3

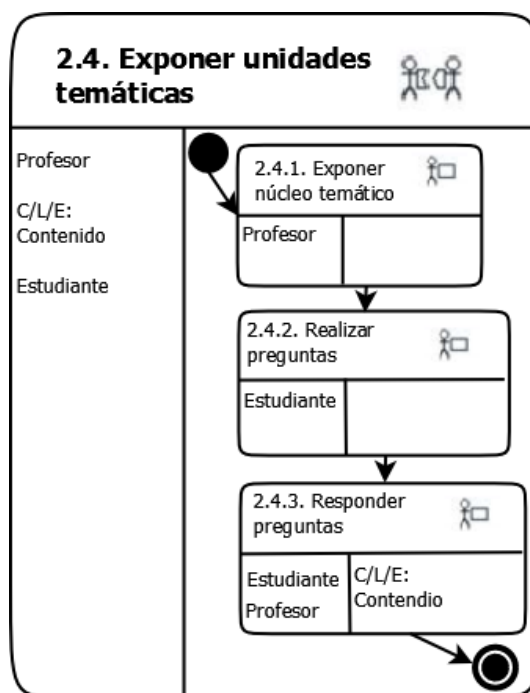


Figura 12. Tarea cooperativa 2.4

En la actividad 2.3.3 (véase **Figura 11**) para elegir el rol adecuado para el Estudiante, el Profesor junto con el Estudiante indagan acerca de los roles propuestos y las Responsabilidades que éstos tienen, eligiendo así el que mejor beneficio tenga tanto para el grupo como para el Estudiante.

El rol elegido influye en el componente o temática sobre la cual posteriormente el Estudiante deberá profundizar en la Tarea 2.5 (véase **Figura 13**) logrando así que el Estudiante se vuelva un experto en su temática, para que cuando regrese a su grupo original (Tarea 2.6), cada uno de los estudiantes esté capacitado para enseñar su temática a los otros y así poder llevar a cabo la tarea 2.7 (véase **Figura 14**).

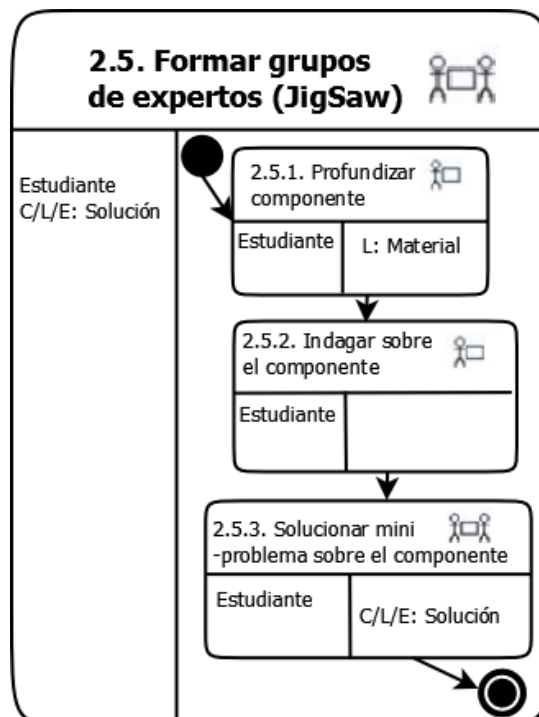


Figura 13. Tarea colaborativa 2.5

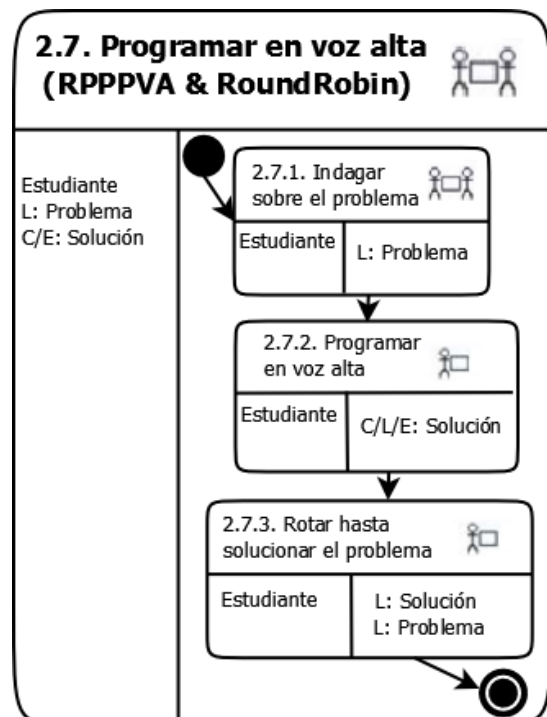


Figura 14. Tarea colaborativa 2.7

En la actividad 2.7.1 (véase **Figura 14**) los estudiantes se preguntan entre si cómo deben solucionar el problema propuesto, a partir del conocimiento que logró cada uno como experto en la tarea 2.5. Después (actividad 2.7.2), un miembro del grupo empieza a programar en voz alta durante un intervalo de tiempo, explicando a sus compañeros (observadores) lo que está haciendo, para que pueda así recibir una retroalimentación de su equipo de trabajo. Posteriormente (actividad 2.7.3), se cambia de programador, es decir, uno de los observadores pasa a programar y el estudiante que estaba programando se vuelve observador, esto se hace con el fin de que todos los estudiantes participen, alcanzando así un conocimiento general del núcleo temático a partir de las explicaciones de sus compañeros y de la retroalimentación recibida por parte de estos. El proceso de rotación se termina cuando se logró llegar a la meta, es decir, se solucione el problema propuesto.

Por último se modelan las tareas 3.1 (véase **Figura 15**), 3.2 (véase **Figura 16**) y 3.3 (véase **Figura 17**).

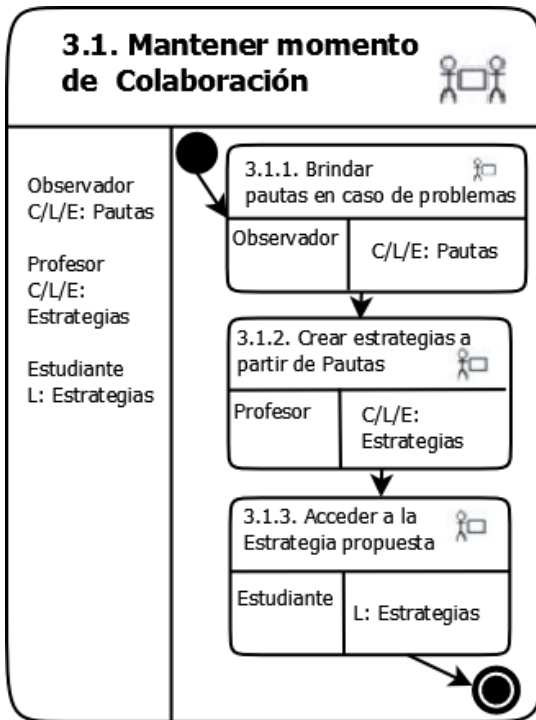


Figura 15. Tarea colaborativa 3.1

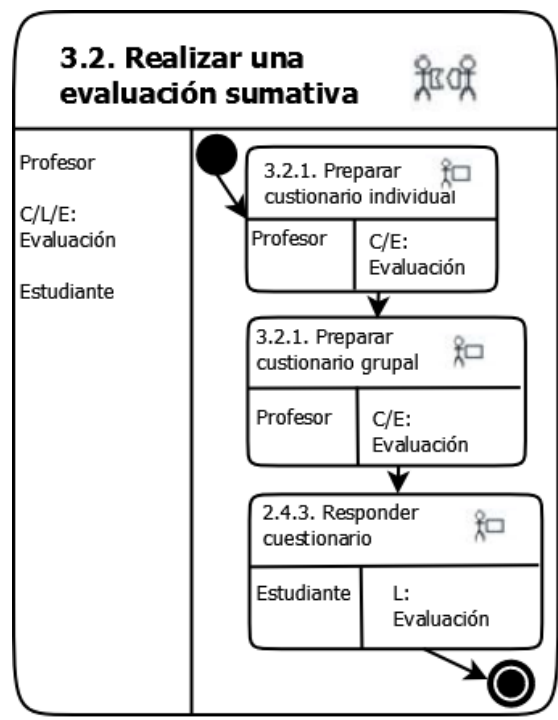


Figura 16. Tarea cooperativa 3.2

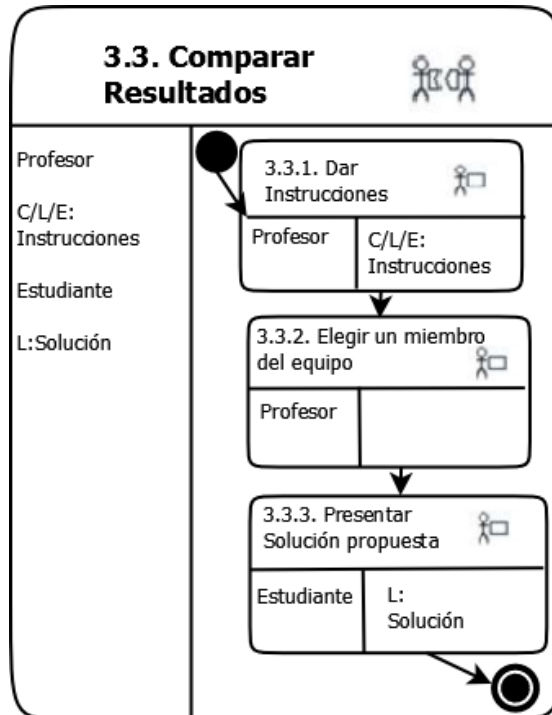
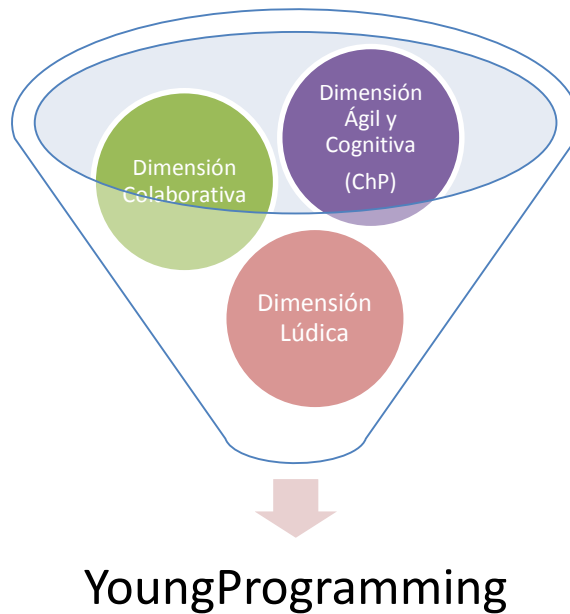


Figura 17. Tarea cooperativa 3.3

Capítulo 4

Este capítulo contiene la descripción del nuevo modelo YoungProgramming considerando la nueva dimensión lúdica y colaborativa, la cual es complementaria a las dos dimensiones: ágil y cognitiva presentes en la versión anterior del modelo (ChildProgramming) considerando la reformulación de algunas prácticas, roles y artefactos para adaptar el modelo a una población conformada por equipos más maduros (jóvenes). La descripción de YoungProgramming parte de una contextualización del modelo propuesto, la definición de su arquitectura, el nuevo proceso de ejecución y su nuevo ciclo de vida.



4.1 CONTEXTO DEL MODELO PROPUESTO

Se denomina YoungProgramming, por la combinación de palabras Young (Jóven), Programming (Programación), “Programación para Jóvenes” y nace de un trabajo preliminar, base inicial de esta investigación, el cual incluye una estrategia integrada para la enseñanza del desarrollo de software en niños que incluye una base conceptual y una metodología de desarrollo de software orientada a niños junto con un entorno de desarrollo integrado para la construcción de software y un programa de formación basado en el aspecto cognitivo, colaborativo y ágil. En la **Figura 18** se representa la arquitectura de esta propuesta llamada ChildProgramming [1].

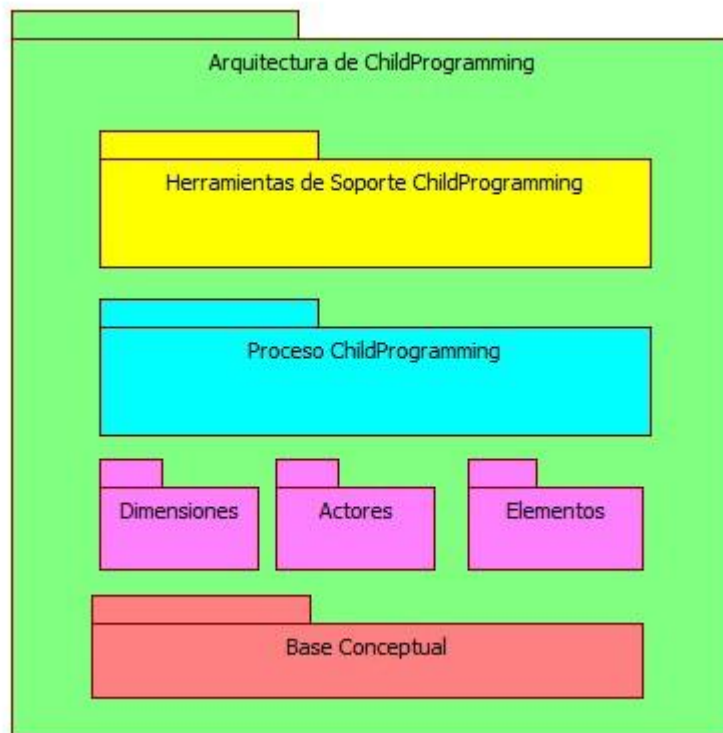


Figura 18. Arquitectura de ChildProgramming [1]. Elaboración propia.

Partiendo de esta base arquitectónica, este trabajo aborda la definición de un proceso de desarrollo basado en la colaboración y la lúdica como ejes centrales y define algunos elementos fundamentales del marco conceptual. En este proceso se hizo necesaria la incorporación de un componente lúdico y colaborativo planteados inicialmente en la propuesta ChildProgramming y además se considera en esta nueva propuesta del modelo, adaptar y reformular las prácticas, roles y artefactos del proceso ChildProgramming orientándolas hacia equipos conformados por jóvenes.

4.1.1 Definición del Proceso YoungProgramming

El nuevo proceso YoungProgramming está orientado a brindar un marco de trabajo para el aprendizaje de la programación en jóvenes, está dividido en tres fases (pre-juego, juego y post-juego) que están compuestas por varias actividades que cubren todo el proceso desde el inicio de la actividad, hasta que se entrega un producto entregable, se analiza, se revisa y se obtienen resultados. El propósito de este proceso es convertir el ambiente de desarrollo

de programas, en un ambiente lúdico y colaborativo para facilitar el aprendizaje. Con la ayuda de la aplicación de la técnica LudiCoP propuesta en el capítulo 3, y aprovechando las mecánicas de juego se puede alcanzar un nivel de motivación tal que los niños mejoren su desempeño en el desarrollo de las actividades.

4.1.2 El rol de LudiCoP en YoungProgramming

El objetivo principal de LudiCoP en este modelo es impactar directamente el proceso de aprendizaje de la programación en jóvenes, enganchando al joven para estimular su autoaprendizaje y el interés por seguir aprendiendo o profundizando en el desarrollo de programas. LudiCoP promueve el aprendizaje tanto individual como del grupo mediante el uso de mecanismos que fomentan los tres principios colaborativos de aprendizaje (Igual participación, Interdependencia positiva y Responsabilidad Individual). Logrando así, que todos y cada uno de los participantes del grupo contribuyan en la solución de un determinado problema, brindando mecanismos para llegar a la solución esperada.

4.1.3 Las Mecánicas de Juego en YoungProgramming

El propósito de las mecánicas de juego en YoungProgramming es crear una serie de experiencias en los jóvenes que aporten una mayor atracción y motivación en el desarrollo de misiones de programación. Las mecánicas de juego se componen de herramientas y técnicas que se utilizan de forma complementaria entre ellos para lograr la consecución de objetivos previamente definidos, alcanzando una alta motivación en los usuarios. Las mecánicas de juego actúan en YoungProgramming como reglas que se deben seguir en el desarrollo de la actividad de tal forma que dicha actividad se asimile a un juego o a una actividad lúdica.

4.2 ARQUITECTURA CONCEPTUAL DE YOUNGPROGRAMMING

La **Figura 19** presenta el modelo arquitectónico tipo módulos de YoungProgramming. Dicho modelo está basado en el modelo arquitectónico ChildProgramming [1]. Este modelo está organizado en 6 paquetes de los cuales dos son prácticamente nuevos y el resto de paquetes están definidos tal cual lo realiza ChildProgramming. El sub-paquete Dimensiones dentro del paquete Componentes Young-Programming ha sido modificado, ya que la dimensión colaborativa propuesta inicialmente se reemplazó por la técnica LudiCoP propuesta en el capítulo 3 de este trabajo de investigación. Además, a los componentes de YoungProgramming se le asocia una Dimensión Lúdica, la cual contiene las mecánicas de juego necesarias para lograr una mayor motivación por parte del Estudiante.

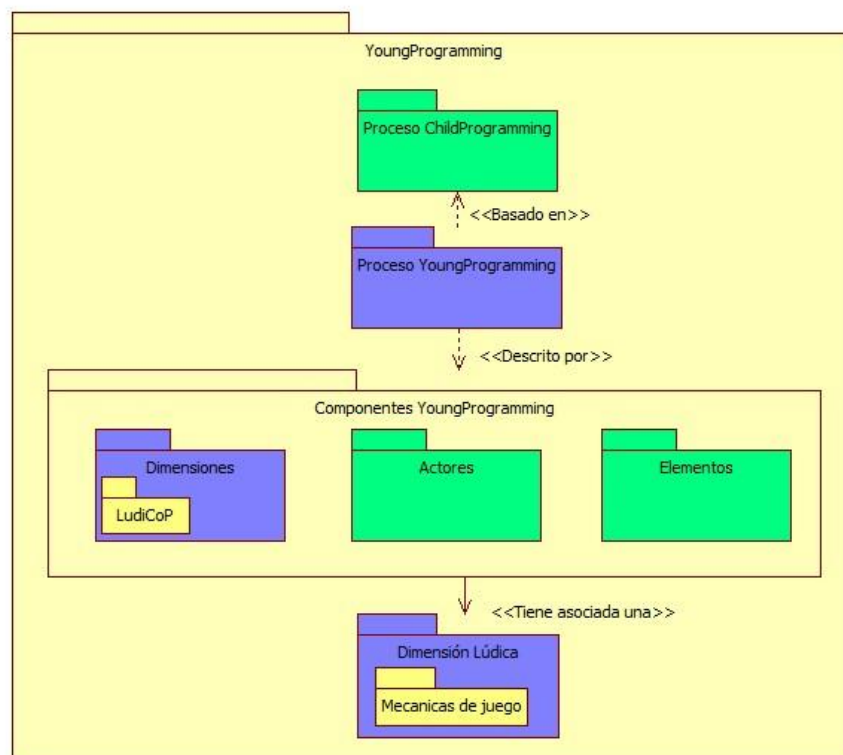


Figura 19. Arquitectura de YoungProgramming.

4.2.1 Actores

YoungProgramming no incluye más actores al modelo original, se mantienen los mismos actores de ChildProgramming, con una breve variación en el rol Estudiante, ya que dentro de LudiCoP el Estudiante se convierte en un Experto, el cual se encarga de transmitir el conocimiento a los demás miembros de su grupo.

Profesor (Tutor): encargado de transmitir el conocimiento teórico y definir las actividades prácticas que realizarán los jóvenes en sus grupos de trabajo durante el transcurso del proyecto.

Estudiantes o Alumnos (Jóvenes): participantes fundamentales, fuente principal de información y ejecutores de las actividades propuestas por el profesor.

Estudiante “Experto”: encargado de transmitir un conocimiento específico a los demás miembros de su grupo.

Observadores (Investigadores): Personas encargadas de apoyar el proceso en cualquier momento teniendo en cuenta las disposiciones del profesor y observando de manera detallada el desarrollo de los niños en sus grupos de trabajo para cada actividad propuesta.

4.2.2 Roles

YoungProgramming no incluye más roles al modelo anterior, pero se agregan tareas a los roles Profesor o Tutor, las cuales tienen que ver con preparar el entorno colaborativo y el entorno lúdico. También se modifican brevemente las tareas del Estudiante o Alumno. Los

anteriores roles se modificaron teniendo en cuenta el contexto de trabajo desarrollado por los jóvenes y por el tutor.

Roles adaptados por YoungProgramming
Internos del Proceso
Profesor
Guía del Equipo(Líder)
Equipo de trabajo
Externos del Proceso
Investigador/Observador externo

Tabla 14. Comparación de Roles de YoungProgramming

A continuación se describen los roles propuestos y actualizados en YoungProgramming:

- **Internos del Proceso**

Profesor: es el responsable del proyecto de aprendizaje en el aula de clase, debe entregar a los equipos las pautas para la realización de la actividad, debe ser el encargado de entregar la misión a realizar, entrenar y monitorizar la metodología de trabajo y estar dispuesto a esclarecer dudas e inquietudes en cualquier momento del desarrollo. El profesor interviene en el proceso siempre, aunque no debe ser intrusivo en el trabajo del equipo. Además es el encargado de tomar las decisiones finales con respecto a la Misión, es quien define y determina los objetivos y requisitos teniendo en cuenta la temática a tratar. Al profesor se agregan otras tareas que están basadas en preparar el entorno colaborativo y lúdico en el aula de clase y comprende las actividades de elegir y mostrar las mecánicas y dinámicas de juego a utilizar.

Guía del grupo: es un aprendiz responsable de asegurar el desarrollo de la Misión acorde a las características y los requisitos de la misma. Debe estar pendiente de que el equipo esté trabajando acorde a las prácticas, valores y reglas establecidas y que se avance en el desarrollo del trabajo según lo previsto. Trabaja igual que el resto del equipo y también es responsable de eliminar las dificultades y de mantener un ritmo productivo como sea posible. El guía del equipo es escogido en consenso por los integrantes del mismo.

Grupo de Trabajo: el grupo de trabajo es el conformado por los jóvenes, quienes tienen la responsabilidad de organizarse para alcanzar las metas propuestas para la Misión y realizar cada tarea acorde a las características y especificaciones de las mismas. El grupo de trabajo está implicado en la valoración del esfuerzo de sus integrantes, la priorización de cada tarea, la revisión de cada uno de sus resultados, mantener continua comunicación entre sí y sugerir soluciones o propuestas en cualquier momento específicamente en momento donde se presenten dificultades e inconvenientes en la realización de la Misión en las reuniones de trabajo continuas.

- **Externos al Proceso**

Investigador u Observador externo: es quien participa de la actividad observando el desarrollo de la misma, inicialmente con el profesor se encargan de describir la Misión y definir los objetivos de la misma, con el fin de determinar el objeto de la observación para la investigación a realizar. Además, participa en las actividades de preparar el entorno colaborativo y lúdico en el aula de clase asesorando al profesor en que mecánicas y dinámicas de juego utilizar y en las reglas de juego.

4.2.3 Mecánicas de Juego Dimensión Lúdica

Las mecánicas de juego que se estudiaron y se analizaron en este trabajo fueron:

Nombre de la mecánica: Puntos	
Descripción:	Es la mecánica de juego más utilizada en la gamificación. Sirve para recompensar a los jóvenes por su buen comportamiento u objetivos alcanzados en el aprendizaje.
Objetivo:	Sirven para llevar la cuenta de la participación positiva del joven en las actividades, determinar sus niveles, desbloquear recompensas y determinar quién va ganando. Es recomendable no dar puntos negativos.
Utilización en YnPg:	Se pueden dar puntos por: <ul style="list-style-type: none"> - Participación en las actividades de aprendizaje. - Completar tareas. - Terminar en primeras posiciones.
Mecánicas asociadas	<ul style="list-style-type: none"> - Recompensas - Tabla de clasificación
Dinámicas generadas:	<ul style="list-style-type: none"> - Competencia entre los jóvenes al observar sus puntos en la tabla de clasificación - Recompensas - Logros

Tabla 15. Puntos en la dimensión lúdica YoungProgramming. Tomado de [69].

Nombre de la mecánica: Desafíos	
Descripción:	Permite a los usuarios competir en una actividad adicional para alcanzar algún beneficio como puntos.
Objetivo:	Sirven para llevar la cuenta de la participación positiva del estudiante en las actividades, desbloquear recompensas y determinar quién va ganando.
Utilización en YnPg:	Se puede utilizar desafíos en caso de un empate en puntuación o en caso de que algún grupo este atrasado en puntaje, se le puede dar la oportunidad de lograr puntos adicionales y estar a la par de sus compañeros para que no pierda la motivación en la actividad.
Mecánicas asociadas	<ul style="list-style-type: none"> - Puntos - Recompensas
Dinámicas generadas:	- Logro al llevar a cabo la consecución del desafío

Tabla 16. Desafíos en la dimensión lúdica YoungProgramming. Tomado de [69].

Nombre de la mecánica: Tablas de Clasificación	
Descripción:	Es el indicador estrella de las actividades lúdicas y busca dar información acerca de la puntuación de los usuarios con respecto a lo demás participantes.
Objetivo:	Busca proporcionar deseo de aspiración y aportar una comparativa entre grupos de jóvenes que lleve a una visión general del desarrollo de las actividades y de los estados de los grupos de jóvenes.
Utilización en YnPg:	Se recomienda visualizar la tabla de clasificación desde la primera actividad de las misiones, hasta culminar todo el proceso, para que los jóvenes puedan ver sus avances en cada sesión y así comparar su puntaje con sus demás compañeros.
Mecánicas asociadas	- Puntos - Tablas de clasificación
Dinámicas generadas:	- Competencia - Estatus

Tabla 17. Tablas de Clasificación en la dimensión lúdica YoungProgramming. Tomado de [69].

Nombre de la mecánica: Beneficios	
Descripción:	Son mecánicas que van a dar al usuario motivación por medio de un premio por su participación positiva dentro del desarrollo de la actividad, puede ser tangible o virtual.
Objetivo:	Busca brindar al usuario una recompensa por su trabajo y desempeño durante el desarrollo de la actividad.
Utilización en YnPg:	Los beneficios que se van a usar en YnPg serán medallas de participación, de expertos en determinados temas. Además, se entregarán certificados a los estudiantes. Se recomienda tener categorías para las medallas para dar motivación a los jóvenes.
Mecánicas asociadas	- Puntos - Recompensas
Dinámicas generadas:	- Competencia - Auto-expresión

Tabla 18. Beneficios en la dimensión lúdica YoungProgramming. Tomado de [69].

Nombre de la mecánica: Tiempos Regresivos	
Descripción:	La cuenta regresiva impulsa al usuario a completar una acción o tareas en un determinado tiempo.
Objetivo:	Su objetivo es motivar al participante a competir contra el reloj y determinar un fin en tiempo para una actividad.
Utilización en YnPg:	La cuenta regresiva se utiliza dando un tiempo considerable para una actividad, de tal manera que sea finita y que el participante la pueda observar para poder distribuir no solo su tiempo sino la estrategia de desarrollo.

Mecánicas asociadas	- Desafíos
Dinámicas generadas:	- Competencia

Tabla 19. Tiempos regresivos en la dimensión lúdica YoungProgramming. Tomado de [69].

4.3 PROCESO YOUNGPROGRAMMING

El proceso YoungProgramming está basado en la aplicación de prácticas ágiles, colaborativas y cognitivas exploradas en el proyecto ChildProgramming, además esta actualizado con actividades de proceso adicionales que hacen parte del componente lúdico agregado al modelo y además se actualizan dos prácticas o dimensiones del modelo anterior (ChildProgramming). Las nuevas actividades dan un marco de trabajo para la utilización de la lúdica y la colaboración en el nuevo modelo, éstas van a ser evaluadas empíricamente con dos estudios de caso, los cuales estarán detallados en el capítulo 5.

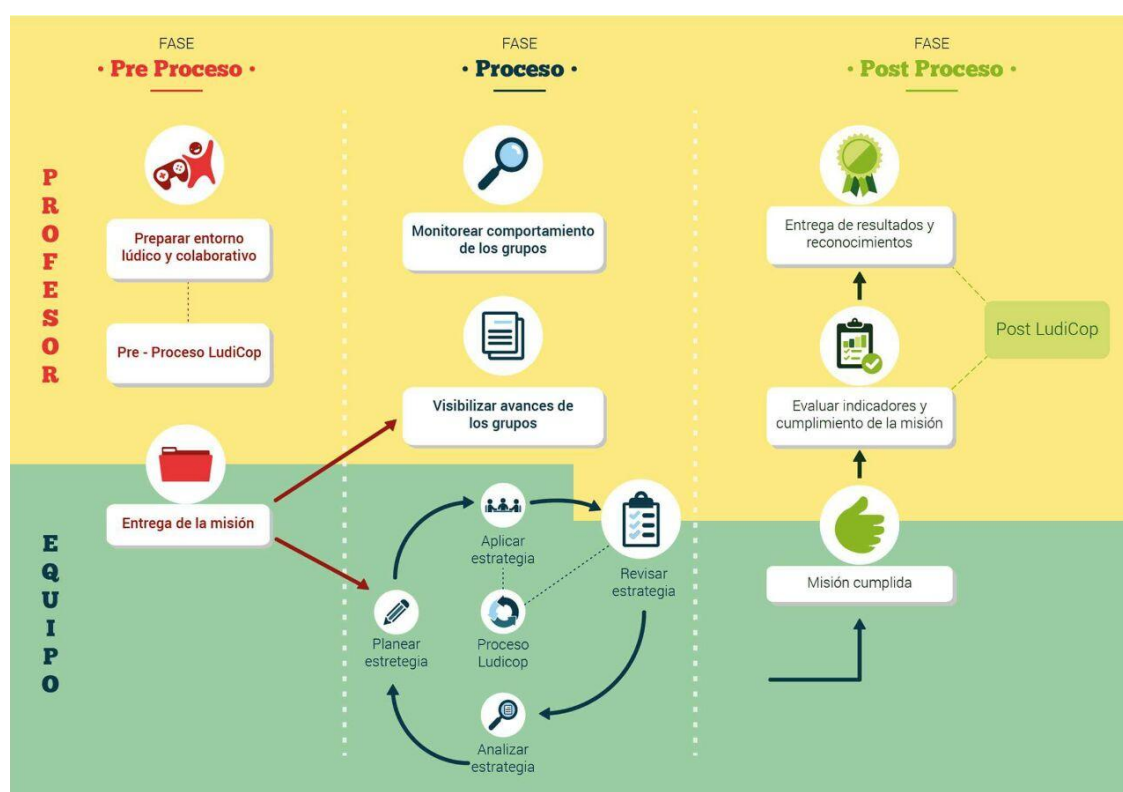


Figura 20. Ciclo de Vida del Proceso YoungProgramming.

4.3.1 Ciclo de vida del proceso YoungProgramming

En la **Figura 20** donde se define el ciclo de vida del proceso YoungProgramming, se puede observar que está dividido en dos subprocesos separados por una línea: En la parte superior muestra las actividades del modelo anterior en color verde, a excepción de la actividad “Entregar Misión” que fue actualizada por la actividad “Establecimiento y entrega de la misión”. En la parte superior en color amarillo están las nuevas actividades del modelo YoungProgramming que corresponden a la planeación y diseño de la lúdica y colaboración por parte del profesor. La actividad “Establecimiento y entrega de la misión” que hace parte de la fase de PRE-PROCESO se encuentra en la línea divisoria, porque ésta requiere la participación del profesor junto con los equipos de jóvenes. Las actividades “Analizar

estrategia” y “Revisar estrategia” que hacen parte de la fase de JUEGO fueron actualizadas para el proceso YoungProgramming del presente trabajo. Las actividades colaborativas y lúdicas que fueron actualizadas ya se encuentran contempladas bajo el marco de LudiCoP, esto se puede ver en las actividades “Preparar entorno lúdico y colaborativo” las cuales ya se encuentran contempladas en la fase de pre-proceso de LudiCoP con sub-actividades como “Describir brevemente la actividad de aprendizaje”, “Definir reglas” y “Definir recompensas”.

El flujo del proceso YoungProgramming puede tomar dos caminos que en ciertos momentos se realizan a la vez y que uno es para el tutor y el otro para los grupos de estudiantes:

Ruta del Profesor: El proceso comienza cuando se realiza la actividad “Preparar entorno lúdico y colaborativo”, luego se procede al “Establecimiento y entrega de la misión”. En este punto pueden iniciar las actividades de los grupos de estudiantes por el otro lado. Continuando con las actividades del profesor, él podrá “Monitorear comportamiento de los grupos” y “Visualizar avances de los grupos”, las cuales se podrán realizar al mismo tiempo. Luego podrá “Evaluar indicadores y cumplimiento de la misión” cuando los grupos de estudiantes entreguen la misión y a continuación podrá realizar la “Entrega de resultados y reconocimientos” que es la parte final del proceso YoungProgramming.

Ruta de los grupos de estudiantes (Workgroup): El proceso inicia cuando el profesor realiza la actividad de “Establecimiento y entrega de la misión” entonces comienzan las actividades de los grupos de estudiantes (Workgroup). Las actividades que realizarán serán “Planear estrategia”, “Aplicar estrategia”, “Revisar estrategia” y “Analizar estrategia” con el fin de cumplir la misión entregada por el profesor, estarán bajo el marco de la fase de proceso de LudiCoP. Después de realizar las actividades el número de veces necesario para completar la misión, entonces se entregará la misión realizada al tutor y éste continuará con las actividades que le corresponde realizar.

A continuación se describirán mejor las nuevas actividades y las actividades tomadas del proceso ChildProgramming, pero que fueron actualizadas para agregarlas al proceso YoungProgramming:

4.3.2 Fase pre-proceso

El propósito de esta fase es preparar el entorno lúdico y colaborativo, y además, establecer y entregar el objetivo de la misión. Preparar el entorno es una nueva actividad que da el mayor soporte a YoungProgramming y de la cual van a depender las otras actividades agregadas al modelo.

Preparar el entorno lúdico y colaborativo – Actividad nueva. Tomado de [69]

Para preparar el entorno hay que realizar varias tareas que son: “Establecimiento de mecánicas de juego” que se van a usar en el desarrollo de las actividades de trabajo en los grupos de jóvenes, Definir las reglas de juego y Definir las recompensas. Con estas tareas y con la entrega de la misión se provee un punto de partida para el aula de clase y las tareas que van a realizar los grupos de jóvenes.

- Establecimiento de mecánicas de juego

En esta actividad el profesor debe definir qué mecánicas de juego utilizará durante el desarrollo de la misión por parte de los grupos de jóvenes. En este trabajo se hizo una evaluación de las mecánicas de juego que tienen mayor impacto en los estudiantes (véase **Sección 4.2.3**), pero el profesor puede utilizar las mecánicas de juego que desee.

- Definir las reglas de Juego

En esta actividad el profesor deberá definir las reglas de juego que se utilizarán durante el desarrollo de la misión con los grupos de estudiantes. El profesor podrá agregar cualquier cantidad de reglas de juego, pero en este caso para YoungProgramming se agregaron ciertas reglas definidas a continuación:

Para uso de logros: Logros como los diplomas, reconocimientos, títulos, etc.

Para uso de recompensas: Definirá qué recompensas dar a los estudiantes, cómo, cuándo y por qué las dará. Como ejemplo se tienen los puntos los cuales pueden ser entregados según los objetivos que cumplan los equipos. Se pueden tener varias estrategias para entregarlos, esto ya depende del profesor. Estrategias como en el caso de si cumple una tarea entonces se le dará una cantidad de puntos.

Establecimiento y entrega de la misión – Actividad actualizada. Tomado de [69]

Esta actividad fue actualizada para YoungProgramming y conserva la base del anterior modelo: el tutor define la misión que los grupos de niños desarrollarán, aclarando el cambio de niños por jóvenes, la cual es la población objetivo del nuevo modelo. Esta misión tendrá los objetivos y metas que los grupos deberán alcanzar, y además el profesor deberá entregar el material necesario en caso de necesitarse para cumplir los objetivos.

La actualización a la actividad es que las tareas que se podrán incluir en esta actividad pueden ser de tipo taller de desarrollo de programas o también puede incluir otras tareas como encuestas o “mini-pruebas” que van a servir de apoyo tanto en el proceso de aprendizaje de la programación como en el de la motivación en los jóvenes.

4.3.3 Fase proceso

En esta fase las iteraciones (Rondas) del modelo anterior (ChildProgramming) continúan igual con el mismo propósito u objetivo: cumplir con la misión y dejar listo el entregable que garantiza el cumplimiento de los objetivos propuestos para la actividad.

La ronda se sigue describiendo igual que en ChildProgramming y continúa incluyendo cuatro Estaciones a seguir, las cuales son: Planear la Estrategia, Aplicar la Estrategia, Revisar la Estrategia y Analizar la Estrategia. Las estaciones Revisar la Estrategia, Aplicar la Estrategia y Analizar la Estrategia se actualizan y la otra estación (Planear Estrategia) se deja como está para el nuevo modelo. Estas estaciones se apoyan en las sub-actividades contenidas dentro de la fase de proceso de LudiCoP, donde en la estación “Aplicar la estrategia”, por ejemplo, será aplicada con el apoyo de la actividad “Programación en voz alta y RoundRobin” propuestos en LudiCoP.

En esta fase de YoungProgramming se incluyen otras actividades, las cuales son “Monitorear el comportamiento de los equipos” y “Visualizar actividades”

Monitorear el comportamiento de los equipos – Actividad nueva. Tomado de [69]

El monitoreo de cómo se están comportando los equipos tienen como fin la toma de decisiones en ciertos casos puntuales. Por ejemplo, cuando algún grupo no está puntuando o no está alcanzando los objetivos propuestos por el profesor, entonces este debe ayudar a que el grupo no se desmotive y continúe. Para ayudar en este aspecto se tienen los puntos adicionales que el profesor dará dependiendo de su criterio y de cómo se ha comportado el

equipo en el cumplimiento de otros aspectos, como por ejemplo los que se plantean en el modelo anterior ChildProgramming (trabajo en pares, trabajo en equipo, organización, etc.).

Visualizar avances de los grupos – Actividad nueva

Mantener informados a los grupos de jóvenes sobre el avance y estado de las mecánicas y dinámicas de juego es importante para que se mantengan concentrados en cumplir con el objetivo y para que el compromiso siempre esté presente. La visualización de las mecánicas de juego mientras realizan las actividades hará que se generen las dinámicas de juego y que el grupo esté enfocado en cumplir los objetivos. Por ejemplo, si se les está mostrando la puntuación en forma de tabla de clasificación ellos estarán comparándose con otros grupos con lo cual se generará competencia entre ellos y se mantendrán entusiasmados para realizar las actividades planteadas por el profesor.

Revisar Estrategia – Actividad actualizada

Esta estación conserva parte de la estación obtenida en el proceso ChildProgramming: Los grupos de jóvenes junto con el Profesor verifican las tareas realizadas y el cumplimiento de las mismas de tal forma que se logre evidenciar en el resultado. Esta estación permite a los integrantes del equipo evaluar el avance de su misión, conocer su ritmo de trabajo, los cuales serán útiles para la planeación de la estrategia en la siguiente ronda.

Hay una modificación que está relacionada con las actividades del profesor: Al estar en la fase de Juego, entonces el profesor debe estar realizando la actividad de “Monitorear comportamiento de los grupos” y aquí en esta estación de “Revisar estrategia” con el equipo, entonces podrá darse cuenta de posibles estrategias buenas o estrategias que están fallando y esto hará más fácil la tarea de monitorización de los equipos, logrando también mejorar el rendimiento del grupo mediante la corrección de los aspectos que están fallando dentro de la estrategia.

Analizar Estrategia – Actividad actualizada

Igual a la anterior estación ésta conserva parte de la estación obtenida en el proceso ChildProgramming: Los grupos de jóvenes y el Guía del Equipo (Líder), evalúa su trabajo como equipo, entorno al desempeño de los integrantes, sus aportes y colaboraciones, así como el compromiso que refleje cada uno, determina si la estrategia empleada funciona o no.

La actualización es igual que en la estación de “Revisar estrategia”, pero solo involucrará a los grupos de estudiantes y no al profesor: Los grupos de estudiantes modificarán o no su estrategia dependiendo de cómo fue su avance y como fue el desempeño de cada integrante del grupo, y estarán pendientes de los puntos y las tablas de clasificación para compararse con los otros grupos de la clase.

Aplicar Estrategia – Actividad actualizada

Esta estación es la más influenciada por parte de LudiCoP, ya que el equipo de trabajo deberá crear una estrategia de tal forma que cada integrante del equipo “programa en voz alta y haga el proceso de rotación” logrando así que todos los miembros del equipo participen.

4.3.4 Fase post-proceso

YoungProgramming conserva el objetivo del anterior modelo, el cual es entregar la Misión Cumplida. La Misión Cumplida corresponde a la solución completamente implementada

acompañada de todo el material asociado al desarrollo de la actividad. El Equipo de Trabajo y el Guía del Equipo entregan la Misión Cumplida al Profesor. El Profesor al cierre de esta Misión, debe evaluar que el trabajo del equipo refleje un proceso de aprendizaje y que haya permitido, de forma efectiva, cumplir con los objetivos de aprendizaje propuestos, pero aquí es donde YoungProgramming hace su aporte e incluye otras actividades que son Evaluar indicadores y cumplimiento de misión, y además , Entrega de resultados y reconocimientos.

Evaluar indicadores y cumplimiento de misión – Actividad nueva. Tomado de [69]

El profesor revisará el cumplimiento de la misión y dependiendo de las mecánicas de juego que utilice, les hará entrega de avances en sus mecánicas de juego o les hará avances en sus dinámicas de juego. Un ejemplo de los avances podría ser en el caso de utilizar la mecánica de juego de puntos. Al evaluar el cumplimiento de la misión podrá entregar ciertos puntos a cada grupo según corresponda. Todo esto irá acompañado con la evaluación de si el objetivo de aprendizaje se cumplió en todos los grupos.

Entrega de resultados y reconocimientos – Actividad nueva. Tomado de [69]

En la entrega de resultados se darán los logros (diplomas, certificados) y si se quiere elementos como las medallas, escarapelas u otros elementos que el profesor haya decidido en las reglas de juego en la fase de pre-proceso, para los estudios de caso se va a utilizar el robot Mbot⁹, el cual podrá ser programado por el equipo con mayor puntuación dentro de la actividad, demostrando así los conocimientos adquiridos durante el proceso. Realizar esta entrega de resultados y de reconocimientos a los grupos de jóvenes es importante para una posible continuación en otras actividades. Esta retroalimentación servirá para ver en qué fallaron, qué les faltó y qué podrían mejorar para una posible actividad siguiente.

⁹ Mbot: kit de robótica educativa para iniciarse en robótica, programación y electrónica basado en Arduino y Scratch. Desarrollado por Makeblock.

Capítulo 5

En el presente capítulo se expone la metodología que se utilizó para este trabajo investigativo realizado en un entorno escolar, incluyendo el contexto de la investigación y la especificación, resultados y análisis de los dos estudios de caso. El primer caso tiene como fin, evaluar la viabilidad de la investigación presentada en los capítulos 3 y 4, en el segundo estudio de caso, se evaluarán los ajustes y mejoras realizadas a la investigación a partir de los resultados del estudio de caso número uno.

5.1 METODOLOGÍA

En [66] plantean que el estudio de caso es una metodología de investigación que estudia un fenómeno contemporáneo en su contexto real, buscando mantener la integridad y las características significativas de los eventos, y es ejecutado cuando el investigador tiene poco control sobre los eventos y cuando los sujetos de estudio son más fáciles de observar en grupo que de manera aislada.

En [67] se considera el estudio de caso como método de investigación apropiado para temas que se consideran prácticamente nuevos. En [68] se indica que el método de estudio de caso es una metodología adecuada para investigar fenómenos en los que se busca dar respuesta a cómo y por qué ocurren los fenómenos, permitiendo estudiar un tema determinado, explorar en forma más profunda y obtener un conocimiento más amplio sobre cada fenómeno, lo cual permite la aparición de nuevas señales sobre los temas que emergen y además juega un papel importante en la investigación.

Esta técnica permite al investigador generalizar desde una instancia concreta a un aspecto más general, ofrecer fuentes de datos de los que se pueden hacer análisis posteriores y así, generar futuros trabajos de investigación y en este caso aportar a partir de experiencias reales una contribución a cambiar prácticas a nivel educativo.

Tal como lo plantea [67] es necesario formular el esquema metodológico de la investigación el cual soportará el trabajo a realizar y permitirá su posterior validación. La validez se va desarrollando a lo largo de todo el estudio, en cada una de sus etapas.

En la **Figura 21** se muestra el procedimiento metodológico utilizado para la investigación.

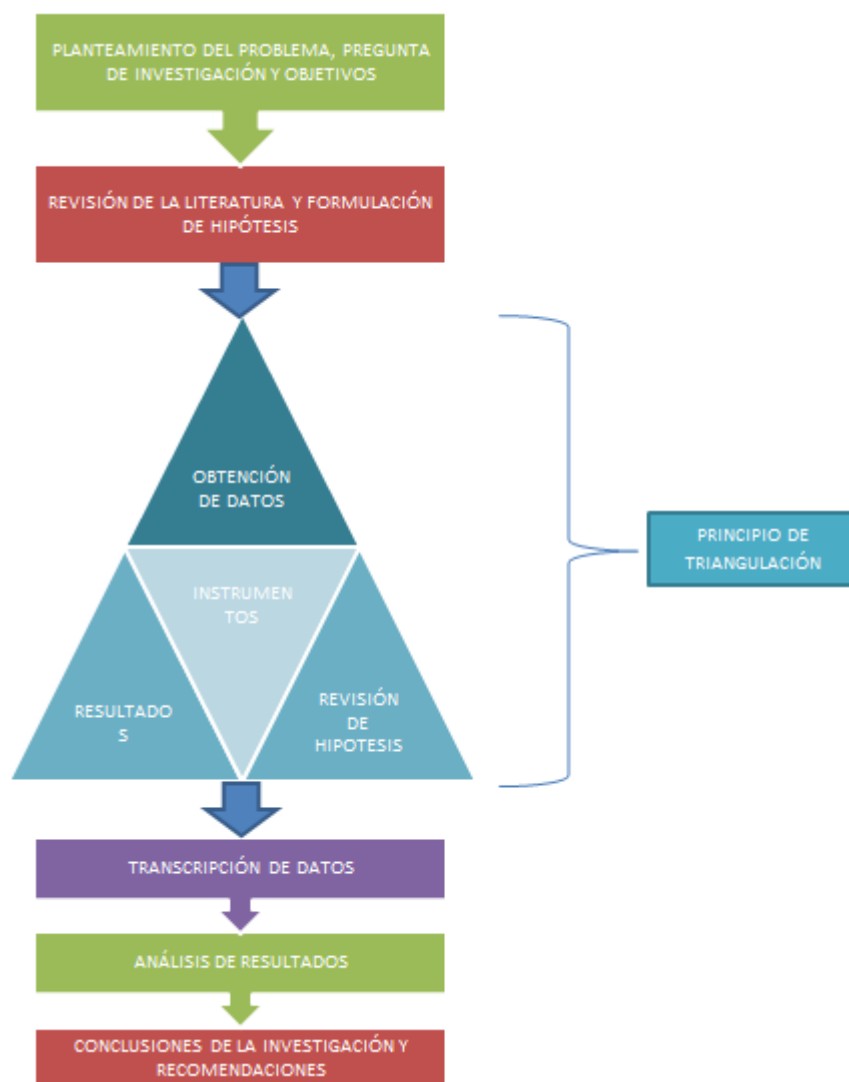


Figura 21. Procedimiento metodológico seguido. Basado en [67].

5.1.1 Instrumentos de Evaluación

Dentro de esta investigación los instrumentos empleados aportan datos que posteriormente son analizados y evaluados para dar validez a la información recogida.

Para esta investigación, los instrumentos seleccionados son:

- **Entrevista:** establece una comunicación interpersonal entre el grupo de investigación y los sujetos de estudio con el fin de obtener datos por escrito de las respuestas de los encuestados y conocer los estados de opinión, características o hechos específicos, que estén relacionados con el tema propuesto. Se considera que este método es más eficaz que el cuestionario porque permite obtener una información más completa y para el caso resulta menos incómodo para los encuestados que son niños poder contestarlas porque les da más libertad de pensar en sus respuestas sin presión alguna.

- **Observación de campo:** este método establece una relación concreta e intensiva entre el equipo de investigación y el hecho social o los actores sociales de los que se obtienen datos que luego se sintetizan para desarrollar la investigación. La observación es un procedimiento de recolección de datos e información que consiste en utilizar los sentidos para observar hechos y realidades sociales presentes y a la gente donde desarrolla normalmente sus actividades.
- **Archivos de datos:** los archivos tienen la responsabilidad de ser fuentes primarias de ilustración y mejoramiento para todos los sectores de la comunidad, en tanto contienen información de documentos que son texto de primera mano, que pueden y deben aprovecharse en el trabajo creador, por parte de investigadores y estudiosos. En esta sección se incluyen test de medición de habilidades y encuestas.
- **Listas de chequeo:** Las listas de chequeo ayudan básicamente a verificar que ciertos aspectos importantes dentro del estudio de investigación se estén revisando y cumpliendo. Es una herramienta apropiada para detectar problemas o defectos que estén ocurriendo en la recolección de la información.
- **Documentos:** Resultantes del trabajo bajo investigación.



Figura 22. Instrumentos de Evaluación

5.2 CONTEXTO DE LA INVESTIGACIÓN

El desarrollo de esta investigación se realizó en las instalaciones de la Institución Educativa Técnico Industrial sede Principal de la ciudad de Popayán ubicada en el barrio "Pomona".

La composición étnico-cultural de la población es homogénea, con mayoría de raza mestiza. Estudiantes procedentes de estratos medio bajo, cuyos padres han superado el nivel de básica primaria. Dentro del factor económico se puede encontrar que la mayoría de las familias tienen un estrato entre 2 y 3, con dedicación en actividades comerciales independientes y empleados permanentes, con un ingreso medio bajo.

La institución cuenta actualmente con el personal administrativo y académico en cabeza del coordinador y el cuerpo de profesores.

En cuanto a recursos, la institución está dotada de los elementos necesarios para el desarrollo de actividades académicas, deportivas y de recreación, dentro de lo cual se destacan los recursos tecnológicos como los habilitados en la sala de informática la cual cuenta con 20 computadores de escritorio, con acceso a internet como parte de un convenio entre la institución, la alcaldía de Popayán y la empresa de telecomunicaciones de Popayán – Emtel.

Para cumplir los objetivos de esta investigación se han diseñado 2 estudios de caso. El primero para evaluar la viabilidad de la investigación realizada en el capítulo 3 y 4 de este trabajo, y el segundo con el fin de refinar la investigación a partir de los hallazgos encontrados en el primer estudio de caso).

5.3 ESTUDIOS DE CASO

5.3.1 ESTUDIO DE CASO 1: APLICANDO YOUNGPROGRAMMING Y LUDICOP EN SU VERSIÓN INICIAL (EC1).

En este estudio de caso se propone explorar y analizar elementos pertenecientes a la colaboración y a la lúdica, que están presentes en YoungProgramming y LudiCoP respectivamente. El estudio de caso se realizó a través de una serie de actividades en diferentes fases, con los jóvenes de grado Once, que darán como resultado la identificación de aspectos importantes en el uso y el impacto del proceso en general. Los estudiantes del grado Once D1 en este caso de estudio serán el grupo experimental y el grado Once D2 será el grupo de control, a quienes se les dio a desarrollar las mismas actividades del otro grupo, pero con la diferencia de que estas actividades fueron desarrolladas bajo el enfoque de aprendizaje tradicional.

Pregunta de investigación

Los elementos pertenecientes a la lúdica que se tendrán en cuenta son los definidos en el capítulo 4 de este trabajo de investigación (véase **Tablas 15 a 19**).

Mientras que los elementos colaborativos que se tendrán en cuenta son los definidos en el capítulo 3 de este trabajo de investigación, bajo el marco de LudiCoP (véase **Tabla 6**).

Para saber el impacto de los anteriores elementos lúdicos y colaborativos en el comportamiento, en la productividad y en la calidad, se realizó una actividad orientada por el modelo YoungProgramming. Este primer estudio de caso buscó resolver la siguiente pregunta:

¿Qué diferencias se presentan en el comportamiento, la productividad y la calidad [69], entre equipos trabajando con YoungProgramming y con el enfoque de aprendizaje tradicional?

Objetivo del estudio

- Analizar el resultado de los dos grupos (control y experimental) en cuanto a comportamiento, productividad y calidad de los productos entregables de los equipo de jóvenes con edades comprendidas entre los 14 y 19 años de edad, de la Institución Educativa Técnico Industrial Sede Principal de Popayán, Cauca.
- Ajustar los componentes lúdicos y colaborativos del modelo, a partir de los resultados obtenidos en este primer estudio de caso.

Selección del estudio

Las actividades de YoungProgramming fueron orientadas específicamente al trabajo con jóvenes los cuales son indicados para poner en práctica y evaluar la propuesta. Para llevar a cabo lo anterior, trabajamos con los grados once de la Institución Técnica Industrial sede Principal, contando con un grupo experimental y otro grupo de control. Por tanto la unidad de análisis para este caso son los equipos de trabajo conformados por 3 jóvenes.

El estudio de caso y los sujetos de investigación.

Este estudio de caso es de tipo embebido [67] donde es considerada como unidad de análisis, jóvenes con edades comprendidas entre los 14 y 19 años, de grado once, quienes para efectos de este caso se constituyen como las fuentes primarias de información, en total son 33 jóvenes divididos en 2 cursos (Once D1 (15) y Once D2(18)). Este estudio de caso tiene un enfoque introductorio pues se introdujo a los jóvenes en la construcción de programas básicos en Scratch y motivar a los jóvenes en la programación en esta herramienta.

Teniendo en cuenta las dinámicas que los jóvenes aplican en sus cursos académicos en la institución, se diseñó el estudio de caso con tres sesiones de trabajo por cada grado Once que fueron desarrolladas desde el 29 de agosto de 2016 hasta el 02 de septiembre de 2016, de la siguiente manera:

Descripción del Caso

Actividad	Descripción	Sesión
Describir brevemente la actividad de aprendizaje	Se brindará a los jóvenes una pequeña introducción acerca del curso que se les va a impartir, se introducirá la herramienta Scratch con fines de exploración y realización de preguntas, además se explicarán los objetivos del presente trabajo, las prácticas de YoungProgramming y una muestra de lo que podrán alcanzar con el curso mediante el Mbot. Esto con el fin de entusiasmar y motivar a los jóvenes con las nuevas clases que van a recibir.	Primera Sesión
Especificar reglas de la actividad	En esta actividad para el grupo experimental se utilizarán las mecánicas de juego de puntos, desafíos, tablas de clasificación, beneficios y tiempos regresivos. Se deben presentar las reglas y restricciones dentro de la actividad para tener un control sobre	Primera Sesión

Actividad	Descripción	Sesión
	comportamientos y circunstancias de la actividad.	
Definir especificación de recompensas comunes para el grupo	Se definirán recompensas que serán otorgadas siempre y cuando el grupo coopere para alcanzar los objetivos comunes de la clase. Para este primer estudio de caso experimental, se entregaran recompensas a los dos primeros equipos que logren cumplir con la misión propuesta.	Primera Sesión
Formar grupos de 3 estudiantes.	Se forman los grupos de tres estudiantes según criterios del docente, garantizando que sean grupos homogéneos en cuanto a niveles de conocimiento.	Segunda Sesión
Identificar al grupo con un nombre, logotipo, slogan	Una vez formados los grupos, los estudiantes deben identificarse mediante un nombre y un slogan, en esta actividad el nombre y slogan más creativo será premiado con puntos antes de iniciar la competencia.	Segunda Sesión
Asignar roles	Se asignarán roles dentro de los grupos de la siguiente manera: un presidente, un vicepresidente y un secretario por cada grupo.	Segunda Sesión
Exponer las unidades temáticas	Se impartirán las nociones básicas de programación, donde aprenderán los conceptos básicos de programación en Scratch.	Segunda Sesión
Formar grupos expertos(JigSaw)	En esta actividad, cada uno de los integrantes de los grupos originales pasará a ser un nuevo integrante de un nuevo grupo “experto” (grupo de presis, vices o secretes) con el objeto de que los alumnos desarrollen sus conocimientos sobre un tema determinado (solucionando un	Segunda Sesión

Actividad	Descripción	Sesión
	problema véase Anexos F, G, H) y que sean capaces de formular métodos eficaces de transmitírselos a otros.	
Retornar a sus grupos originales	Los grupos de “expertos” (presis, vices y secres) se deshacen y los estudiantes pasan de nuevo a sus grupos originales “rompecabezas”, formado cada uno de ellos por alumnos que han llegado a dominar distintos subtemas.	Tercera Sesión
Programar en voz alta (RPPPVA & RoundRobin)	Cada integrante del grupo original programa (en voz alta), de acuerdo al conocimiento adquirido como “experto”, durante 5 minutos, compartiendo así su conocimiento con los demás (solucionando un problema véase Anexo E). Rotando hasta que todos los integrantes del grupo programen (RoundRobin). Este proceso de rotación se acaba cuando se logre solucionar el problema propuesto.	Tercera Sesión
Mantener el momento de colaboración	En esta actividad el docente y los observadores supervisarán cada una de las actividades propuestas y en caso de problemas intervendrán para que los grupos trabajen de forma adecuada.	Todas las sesiones
Realizar una evaluación sumativa	El objetivo de esta actividad es testear el nivel final de conocimientos (véase Anexo I) y verificar el desempeño de ciertos aspectos a nivel grupal (véase Anexo J).	Tercera Sesión
Hacer que los grupos comparen	En esta actividad los diferentes grupos compararán las diferentes soluciones que se han dado para la misión propuesta, con el fin de que los miembros del grupo expliquen a	Tercera Sesión

Actividad	Descripción	Sesión
entre sí sus resultados	los de otros grupos como lograron solucionar la misión, logrando así que reafirmen sus conocimientos.	

Tabla 20. Actividades Estudio de Caso 1 – Guía LudiCoP

Diseño de los Indicadores y mediciones. Tomado de [69]

Para obtener la información necesaria para este Estudio de Caso, particularmente, y dar respuesta a la pregunta de investigación se tomaron un conjunto de mediciones e indicadores de [69]. La tabla 21 muestra un resumen de los indicadores y métricas definidos:

Pregunta	Indicadores	Mediciones	Instrumentos
¿Qué diferencias se presentan en el comportamiento, la productividad y la calidad, entre equipos trabajando con YoungProgramming y con el enfoque de aprendizaje tradicional?	Diferencia entre los niveles de productividad.	Nivel de comportamiento observado en los equipos de trabajo en ambos tipos de equipos.	Observación, Protocolo de Observación (Anexo A), Misión (Anexo E)
	Diferencia entre los niveles de comportamiento.		
	Diferencia entre los niveles de calidad.	Nivel de Productividad observado en los equipos de trabajo en ambos tipos de equipos	Observación, Protocolo de Observación (Anexo B), Misión (Anexo E)
		Nivel de Calidad observado en los equipos de trabajo	Observación, Protocolo de Observación(Anexo C), Reporte de Errores (Anexo D), Misión (Anexo E)

Tabla 21. Tabla de Indicadores y mediciones.

Indicadores

Diferencia entre los niveles de Productividad: Es la diferencia entre la productividad del grupo experimental (Once D1) que realiza actividades y prácticas de YoungProgramming y el grupo de control (Once D2) que realiza las actividades y prácticas bajo el enfoque de aprendizaje tradicional. Cabe resaltar que los niveles de Productividad se encuentran en el rango de 0 a 100 por ciento.

Diferencia entre los niveles de Comportamiento: Es la diferencia entre el comportamiento del grupo experimental (Once D1) que realiza actividades y prácticas de YoungProgramming y

el grupo de control (Once D2) que realiza las actividades y prácticas bajo el enfoque de aprendizaje tradicional. Cabe resaltar que los niveles de Comportamiento se encuentran en el rango de 0 a 100 por ciento.

Diferencia entre los niveles de Calidad: Es la diferencia entre la calidad del grupo experimental (Once D1) que realiza actividades y prácticas de YoungProgramming y el grupo de control (Once D2) que realiza las actividades y prácticas bajo el enfoque de aprendizaje tradicional. Cabe resaltar que los niveles de Calidad se encuentran en el rango de 0 a 100 por ciento.

Mediciones. Tomado de [69].

Comportamiento: Es la forma de actuar o reaccionar un individuo frente a una situación, problema o actividad. Trata de como también las personas u organismos proceden frente a los estímulos en relación con el entorno¹⁰. Esta medición permite medir un conjunto de conductas comunes (ver **Anexo A**) mediante el instrumento de observación (Protocolo de Observación) relacionado con la actividad, que los jóvenes en sus equipos de trabajo exhiben con mayor frecuencia. La fórmula que se ha utilizado para valorar estas conductas está dado por:

$$C = \left\{ \frac{1}{7} \left[(NR_s * 100) + (NR_A * 50) + (NR_I * 10) \right] \right\} \quad (1)$$

Donde C corresponde a comportamiento, NR_s es el número de respuestas satisfactorias, NR_A es el número de respuestas aceptables, NR_I es el número de respuestas insatisfactorias. Cabe resaltar que la forma de diligenciar estas respuestas está contenida en el **Anexo A**.

El factor (1/7) se relaciona con el número de características contenidas en el instrumento de observación (Protocolo de observación). El valor 100, 50 y 10 que operan con el NR_s , NR_A , NR_I corresponden a la valoración dada al número de observaciones con evaluación Satisfactoria, Aceptable e Insatisfactorio respectivamente. Para esta métrica se acepta como valores de conductas adecuadas aquellas que están por arriba del 50%.

Productividad: Se define como la manera acertada en que se han combinado y utilizado los recursos para cumplir con los objetivos específicos deseados, en el tiempo programado¹¹. Esta medición permite medir el cumplimiento, aplicación o evidencia de los equipos en su trabajo y en el producto obtenido en la actividad a través de las características definidas en el instrumento de observación (ver **Anexo B**) (Evaluación de Conformidad del Investigador). Se obtiene a partir de la siguiente fórmula (2):

$$P = 10 * \left[(NR_5 * 100) + (NR_4 * 80) + (NR_3 * 60) + (NR_2 * 40) + (NR_1 * 20) \right] / 100 \quad (2)$$

Donde P corresponde a la productividad, NR_5 es el número de respuestas con valoración 5, NR_4 es el número de respuestas con valoración 4, NR_3 es el número de respuestas con valoración 3, NR_2 es el número de respuestas con valoración 2, NR_1 es el número de respuestas con valoración 1. El valor 100, 80, 60, 40 y 20 que operan con el NR_5, NR_4, NR_3, NR_2 y NR_1 , corresponden a la valoración dada al número de observaciones con evaluación en 1, 2, 3, 4 y 5 respectivamente. Cabe resaltar que la forma de diligenciar estas respuestas está contenida en el **Anexo B**.

Para esta métrica se acepta como valores de adecuados aquellos que estén por encima de 50%.

¹⁰ <http://definicion.de/comportamiento/>

¹¹ <http://tesis.uson.mx/digital/tesis/docs/7268/Capitulo1.pdf>

Calidad: Se define como la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor¹².

Esta medición permite medir la condición, estado o presentación del trabajo mediante el instrumento de Observación (ver **Anexo C**) teniendo en cuenta las cualidades más representativas del producto como tal durante y después de la actividad. Se obtiene a partir de la siguiente fórmula:

$$Cal = \left[\frac{(NR_{SI} + NR_{NO})}{6} \right] * 100 \quad (3)$$

Donde Cal corresponde a Calidad, NRSI es el número de respuestas afirmativas, NRNO es el número de respuestas negativas. El número 6 corresponde al número de características del instrumento de calidad evaluadas a los equipos. Cabe resaltar que la forma de diligenciar estas respuestas está contenida en el **Anexo C**.

Para esta métrica se acepta como valores adecuados aquellos que estén por encima de 50%.

Resultados Cuantitativos

En este estudio de caso se conformaron los grupos de forma equitativa a consideración de los investigadores de la siguiente manera:

Grupo de Control: 5 grupos, cada uno con 3 integrantes conformados por dos (2) hombres y una (1) mujer.

Grupo Experimental: 6 grupos, cada uno con 3 integrantes de los cuales 1 grupo estuvo conformado por 3 mujeres y los 5 grupos restantes estuvo conformado por dos hombres y una mujer.

A continuación se presentan los principales resultados cualitativos del estudio de caso, en términos de productividad, comportamiento y calidad tanto para los equipos experimentales como para los equipos de control.

Productividad Grupo Experimental

Nombre del Equipo	Curs o	Numero de respuestas de valor 5.0	Numero de respuestas de valor 4.0	Numero de respuestas de valor 3.0	Numero de respuestas de valor 2.0	Numero de respuestas de valor 1.0	Resultado Productividad (Fórmula 2)
Wings	11d	7	1	0	1	1	84%
Mix electro	11d	7	2	0	0	0	88%
Iluminado s	11d	4	5	0	0	1	82%
New Girls	11d	9	0	0	0	1	92%

¹² RAE: Real Academia de la Lengua

Nombre del Equipo	Curs o	Numero de respuestas de valor 5.0	Numero de respuestas de valor 4.0	Numero de respuestas de valor 3.0	Numero de respuestas de valor 2.0	Numero de respuestas de valor 1.0	Resultado Productividad (Fórmula 2)
Team puz play	11d	1	2	6	0	1	64%
Octopus	11d	9	0	0	0	1	92%

Tabla 22. Productividad grupo experimental EC1

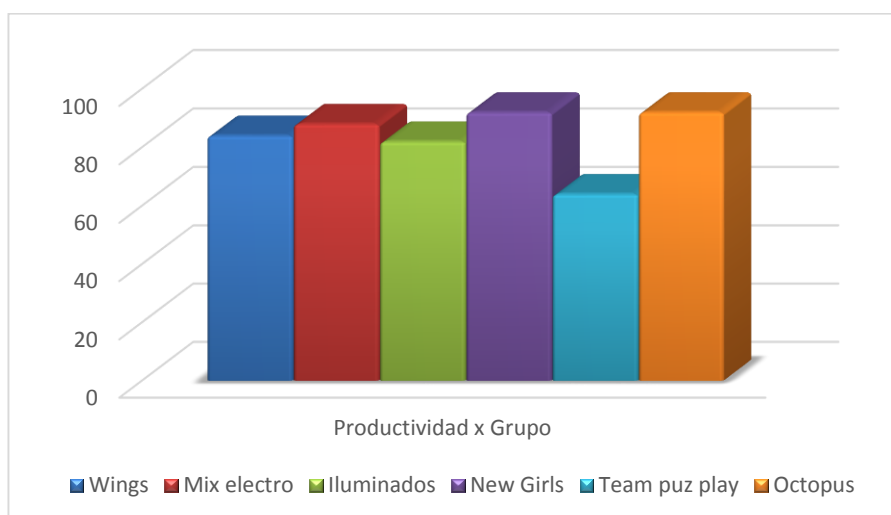


Figura 23. Productividad grupo experimental EC1

Productividad Grupo de Control

Nombre del Equipo	Curso	Numero de respuestas de valor 5.0	Numero de respuestas de valor 4.0	Numero de respuestas de valor 3.0	Numero de respuestas de valor 2.0	Numero de respuestas de valor 1.0	Resultado Productividad (Fórmula 2)
Los Kemoción	11d	0	0	0	0	10	20%
Zuly y sus amigos	11d	1	3	5	1	0	68%
Suicide Squad	11d	1	3	5	1	0	68%
Yi-Li-Ca	11d	1	2	6	1	0	66%
Los Sebachos	11d	1	4	2	3	0	66%

Tabla 23. Productividad grupo de control EC1

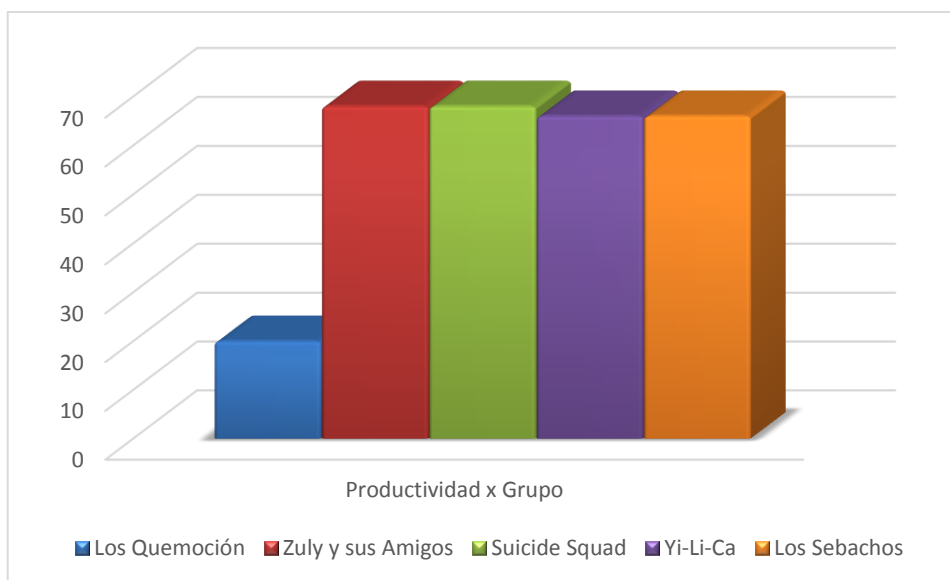


Figura 24. Productividad grupo de control EC1

Calidad Grupo Experimental

Nombre del Equipo	Curso	Numero de respuestas Positivas (Si)	Numero de respuestas Negativas (No)	Resultado Calidad (Fórmula 3)
Wings	11d	4	2	66%
Mix electro	11d	4	2	66%
Iluminados	11d	4	2	66%
New Girls	11d	6	0	100%
Team puz play	11d	3	3	50%
Octopus	11d	6	0	100%

Tabla 24. Calidad grupo experimental EC1

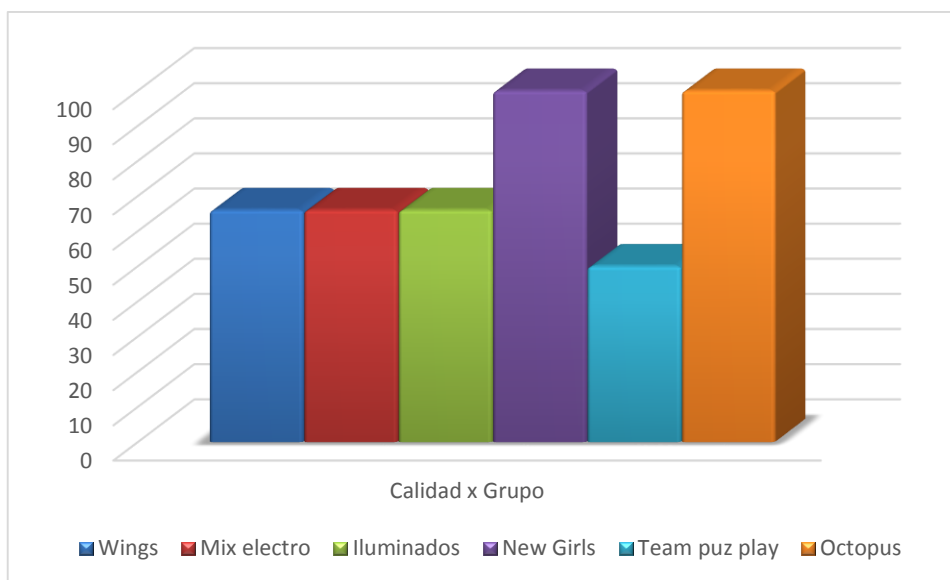


Figura 25. Calidad grupo experimental EC1

Calidad Grupo de Control

Nombre del Equipo	Curso	Numero de respuestas Positivas (Si)	Numero de respuestas Negativas (No)	Resultado Calidad (Fórmula 3)
Los Kemoción	11d	0	6	0%
Zuly y sus amigos	11d	3	3	50%
Suicide Squad	11d	4	2	66%
Yi-Li-Ca	11d	4	2	66%
Los Sebachos	11d	3	3	50%

Tabla 25. Calidad grupo de control EC1

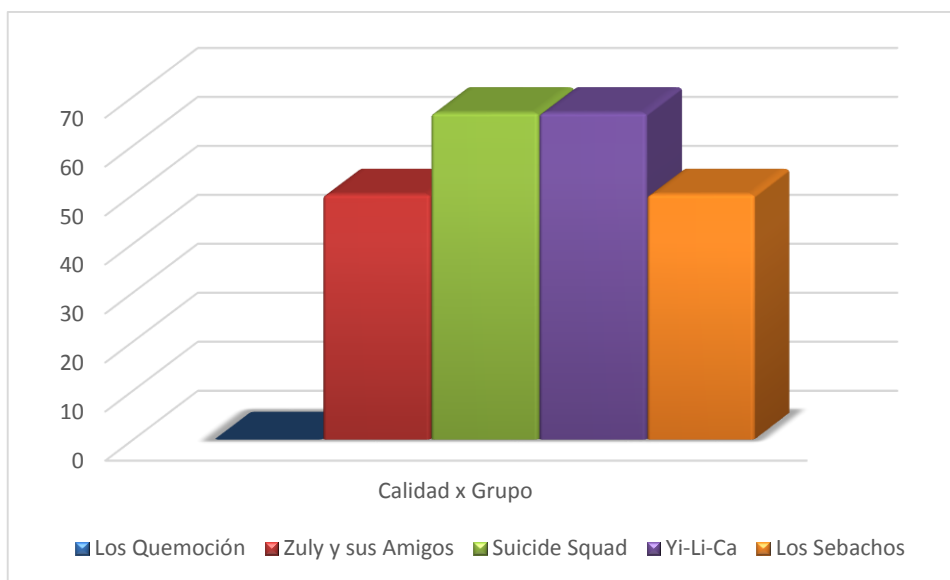


Figura 26. Calidad grupo de control EC1

Comportamiento Grupo Experimental

Nombre del Equipo	Curso	Numero de respuestas satisfactorias	Numero de respuestas aceptables	Numero de respuestas insatisfactorias	Resultado Comportamiento (Fórmula 1)
Wings	11d	6	1	0	92%
Mix electro	11d	5	2	0	85%
Iluminados	11d	4	3	0	78%
New Girls	11d	5	2	0	85%
Team puz play	11d	2	5	0	64%
Octopus	11d	6	1	0	92%

Tabla 26. Comportamiento grupo experimental EC1

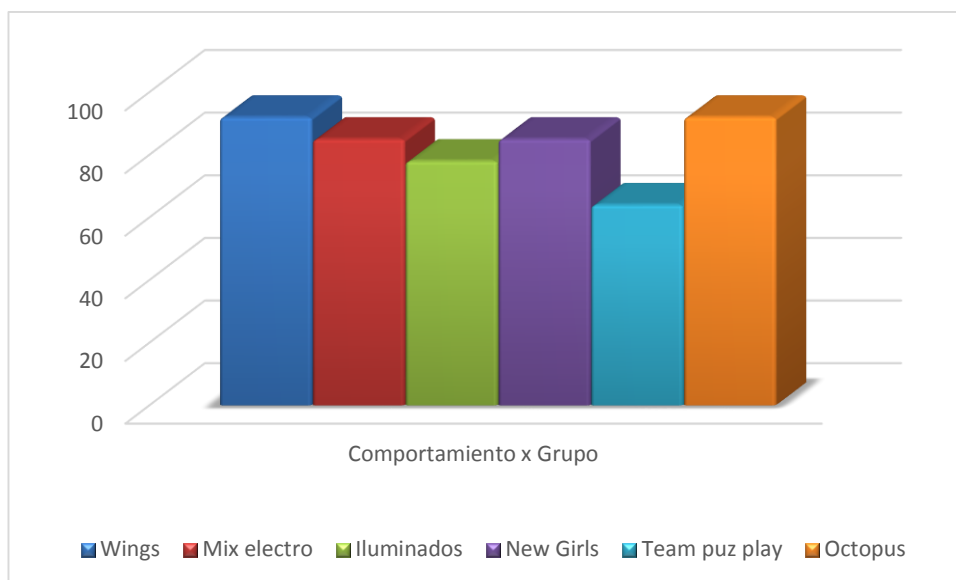


Figura 27. Comportamiento grupo experimental EC1

Comportamiento Grupo de Control

Nombre del Equipo	Curso	Numero de respuestas satisfactorias	Numero de respuestas aceptables	Numero de respuestas insatisfactorias	Resultado Comportamiento (Fórmula 1)
Los Kemoción	11d	0	0	7	10%
Zuly y sus amigos	11d	1	5	1	51%
Suicide Squad	11d	2	5	0	64%
Yi-Li-Ca	11d	1	5	1	51%
Los Sebachos	11d	2	5	0	64%

Tabla 27. Comportamiento grupo de Control EC1

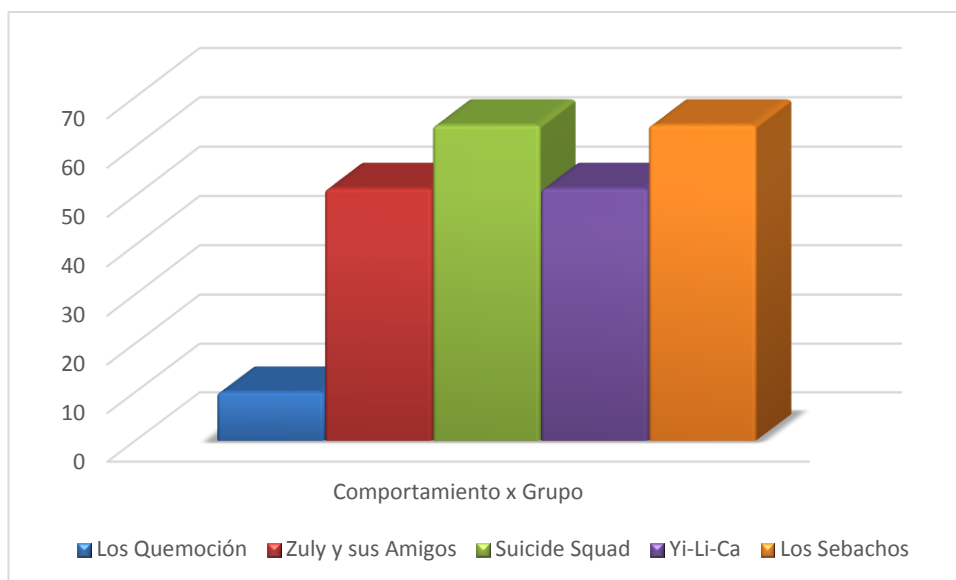


Figura 28. Comportamiento grupo de control EC1

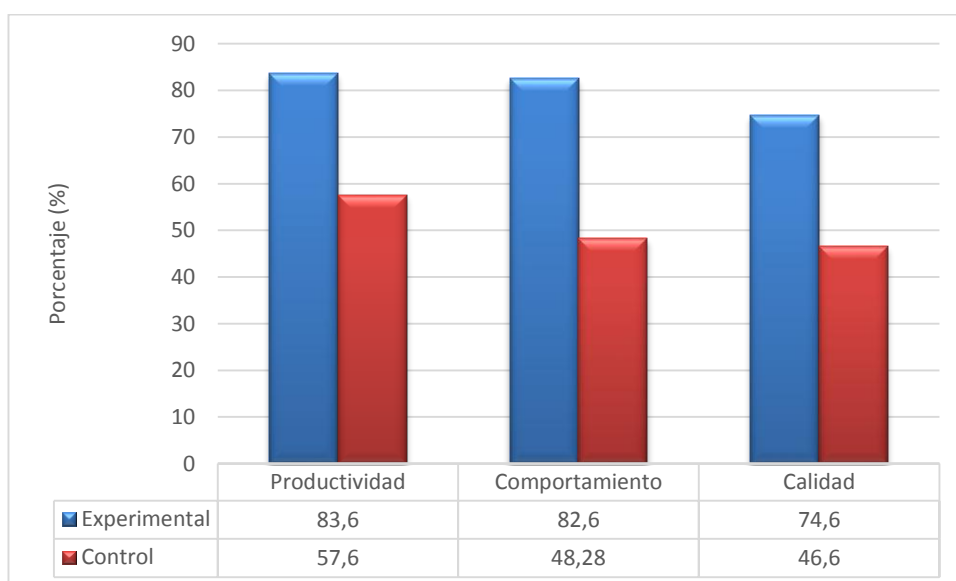


Figura 29. Porcentajes por cursos (Experimental y Control)

En la figura 29 se puede evidenciar la diferencia en cuanto a niveles de productividad, comportamiento y calidad entre los grupos de control y experimental.

Evaluación de Habilidades Adquiridas

En Colombia, la Ley General de Educación en su Artículo 77 otorgó la autonomía escolar a las instituciones en cuanto a libertad para la adopción de métodos de enseñanza y otros aspectos; en la misma perspectiva, con la expedición del Decreto 1290 de 2009, el gobierno nacional otorga la facultad a los establecimientos educativos para definir el SIEE (Sistema Institucional de Evaluación de los Estudiantes), siendo esta una tarea que exige estudio, reflexión y análisis [70].

Los métodos de evaluación de aprendizaje más usados en Colombia son exámenes que miden el aprendizaje y la competencia de los alumnos, los cuales utilizan métodos numéricos de 0 a 5 o de 0 a 10 y se aprueba con 3.0 en adelante y con 6.0 en adelante,

respectivamente; este sistema de calificación inició desde el año 2009 y fue aprobado por el Ministerio de Educación Nacional.

En este estudio de caso se realizó un test de habilidades (véase **Anexo I**) a los grupos Experimental y de Control, con el fin de evaluar el aprendizaje después del uso de la estrategia. Este test, con preguntas tomadas de [71], arrojó como resultado lo siguiente:

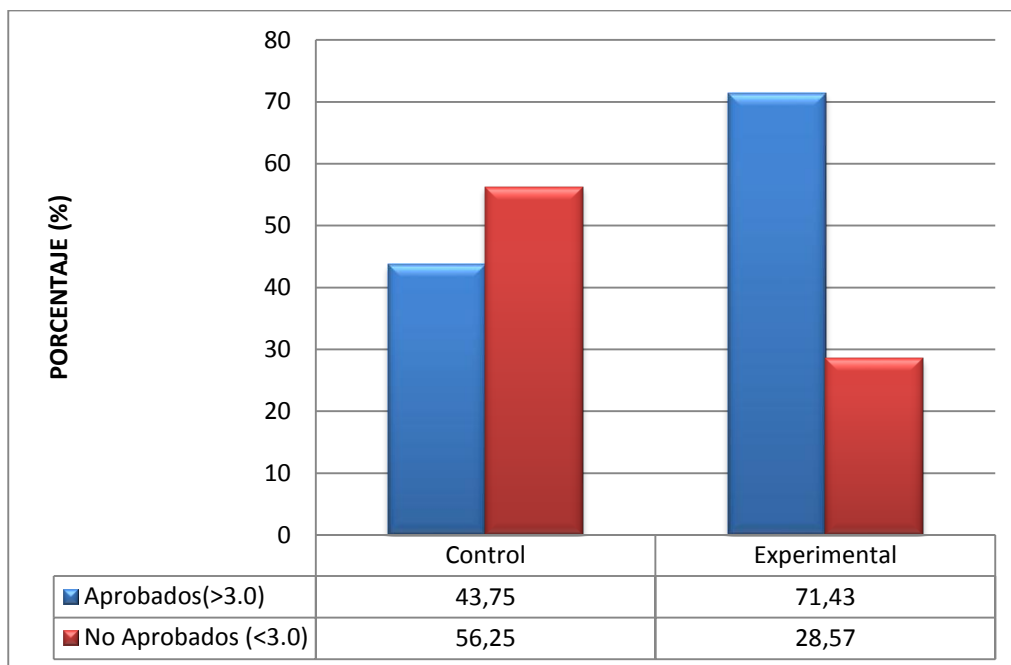


Figura 30. Porcentajes de Evaluación de Aprendizaje por cursos (Experimental y Control)

En la figura 30 se puede evidenciar la diferencia entre la cantidad de estudiantes aprobados en los grupos experimental y de control, mostrando así el efecto positivo del uso de la estrategia propuesta en el aula de clase.

Es importante resaltar el resultado obtenido en las evaluaciones del aprendizaje de las mujeres dentro de los grupos que participaron en la actividad; se notó una gran diferencia entre el grupo de control y el grupo experimental como se puede observar en la **Figura 31**, arrojando como resultado que las mujeres del grupo experimental las cuales utilizaron la estrategia YoungProgramming tuvieron un mejor desempeño y resultado en la realización de los Test.

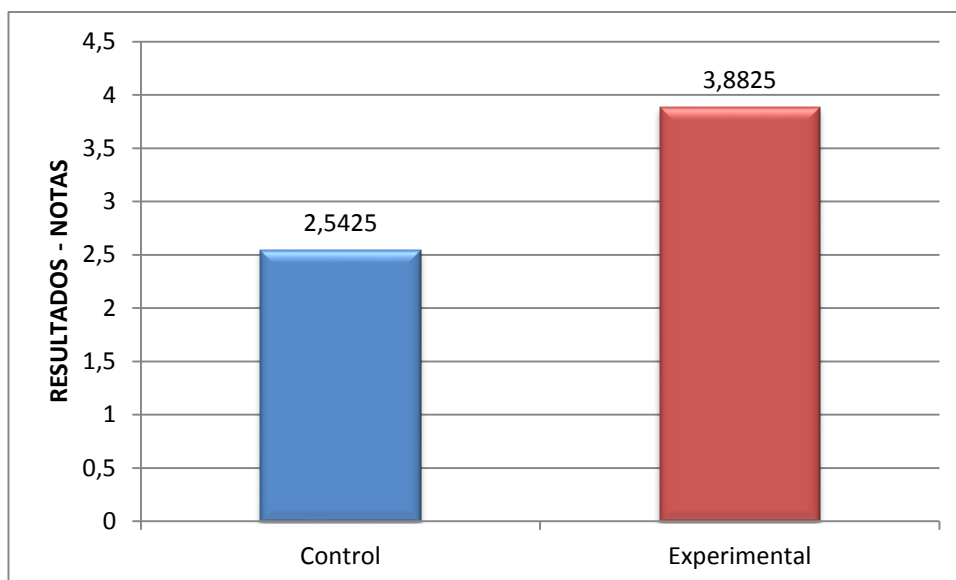


Figura 31. Porcentajes de Evaluación del Aprendizaje de Mujeres por cursos (Experimental y Control)

Análisis de Encuestas

Al finalizar el estudio de caso se realizó una serie de preguntas tipo encuesta (véase **Anexo J**) las cuales fueron respondidas con indicadores de 1 a 5, con el fin de evaluar si los estudiantes percibían la existencia y presencia de las características y artefactos (Responsabilidad individual, Comunicación, Interdependencia Positiva, Igual Participación, Trabajo Colaborativo, Lúdica y Usabilidad) dentro de la estrategia YoungProgramming, como resultado se pudo observar (véase **Figura 32**), que aunque no existió la presencia de estas características en un valor igual a 5, si se percibió entre un 3,7 y un 4,3 la existencia de ellas. En cuanto a usabilidad los alumnos consideraron en un 3,7 que les pareció usable y que sería de agrado para ellos que esta estrategia fuera usada por sus docentes en caso de que ellos les enseñen a programar.

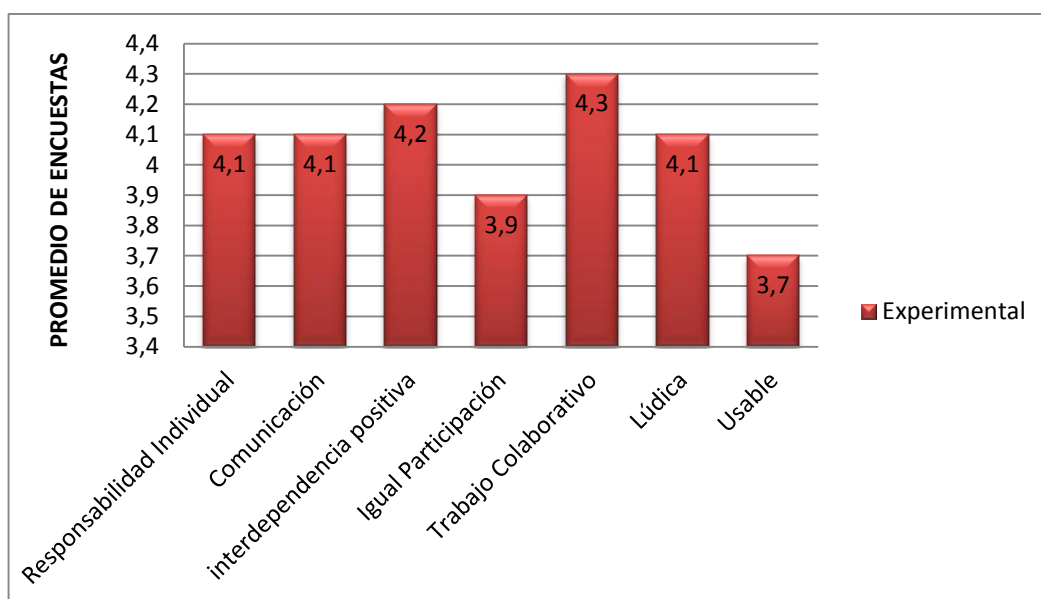


Figura 32. Resultados Encuestas (Experimental).

Análisis de Resultados

Análisis de Resultados Cualitativos

Los indicadores de la estrategia Colaborativa y Lúdica fueron mejores en cuanto a la productividad, comportamiento y calidad. La productividad, comportamiento y calidad tienen mejores resultados en las actividades con la estrategia YoungProgramming (grupo experimental) que con el método de aprendizaje tradicional (grupo de control).

La **Figura 29** (Porcentajes por cursos - Experimental y Control) muestra mejores resultados en los datos del curso experimental en las tres mediciones (productividad, calidad y comportamiento).

En productividad y comportamiento se muestra una alta diferencia, la cual podría estar reflejada por la aplicación de la estrategia YoungProgramming con la incorporación de colaboración y lúdica, debido a que es el componente significativamente diferente entre los dos tipos de equipo. Respecto a la calidad se muestra, de igual forma, una diferencia, aunque no de la misma magnitud, que en los aspectos de productividad y comportamiento, pero que también podría estar reflejada por la aplicación y uso de la estrategia propuesta.

Respecto a las observaciones, se puede analizar que los jóvenes ven en el aula de clase un juego en el que van ganando puntos y otros reconocimientos más que el ambiente de aprendizaje tradicional. Se mezcla el aprender con el jugar y colaborar, al agregar los elementos de lúdica y colaboración en el ambiente de aprendizaje o al mezclar dichos elementos con las actividades pertenecientes a su aprendizaje como tal. Además se puede concluir que la colaboración y la lúdica estuvieron presentes en el desarrollo de esta actividad, como lo concluyeron los jóvenes que participaron en el uso de la estrategia y lo cual se puede observar en la **Figura 32**.

Uno de los resultados importantes encontrados es el desempeño de las mujeres, las cuales obtuvieron mejores resultados en la medición del aprendizaje de la programación con la estrategia Young Programming (grupo experimental) que con el método de aprendizaje tradicional (grupo de control) en cuanto a medición del aprendizaje después de la actividad (véase **Figura 31**) de igual manera los resultados generales del aprendizaje de la programación en el grupo en el cual se aplicó la estrategia YoungProgramming (grupo experimental), en el cual se puede observar que los resultados son notorios y muy positivos (véase **Figura 29**), tuvo una gran diferencia respecto al grupo en el cual se utilizó el método de aprendizaje tradicional (grupo de control).

Cabe resaltar que el resultado de la aplicación de esta estrategia se vio reflejado, gracias a la conformación de los grupos de jóvenes que trabajaron en la actividad, la cual fue de forma no independiente; con esto se puede deducir que la colaboración involucra en la primera fase una estructuración, quizá a futuro cuando los estudiantes ya entiendan este aspecto se puede obviar esta selección y darles la oportunidad de conformar sus grupos de forma autónoma.

5.3.2 ESTUDIO DE CASO 2: APLICANDO YOUNGPROGRAMMING Y LUDICOP EN UNA VERSION MEJORADA (EC2).

En este estudio de caso se propone aplicar YoungProgramming y LudiCoP en su versión mejorada. El estudio de caso se realizó a través de una serie de actividades en diferentes fases, con los jóvenes de grado Décimo, que darán como resultado la identificación de aspectos importantes en el uso y el impacto del proceso en general. Los estudiantes del

grado Décimo D en este caso de estudio serán la unidad de análisis [67], a quienes se les dio a desarrollar actividades bajo el enfoque de YoungProgramming y LudiCoP.

Pregunta de investigación

Los elementos pertenecientes a la lúdica que se tendrán en cuenta son los definidos en el capítulo 4 de este trabajo de investigación (véase **Tablas 15 a 19**).

Mientras que los elementos colaborativos que se tendrán en cuenta son los definidos en el capítulo 3 de este trabajo de investigación, bajo el marco de LudiCoP (véase **Tabla 6**).

Para saber el impacto de los anteriores elementos lúdicos y colaborativos en el comportamiento, en la productividad y en la calidad, se realizó una actividad orientada por el modelo YoungProgramming. Este primer estudio de caso buscó resolver la siguiente pregunta:

¿Qué resultados se presentan en el comportamiento, la productividad y la calidad [69], en equipos trabajando con YoungProgramming en su versión refinada?

Objetivo del estudio

- Analizar el resultado de la unidad de análisis en cuanto a comportamiento, productividad y calidad de los productos entregables de los equipos de jóvenes con edades comprendidas entre los 14 y 19 años de edad, de la Institución Educativa Técnico Industrial Sede Principal de Popayán, Cauca.
- Ajustar los componentes lúdicos y colaborativos del modelo, a partir de los resultados obtenidos en este segundo estudio de caso.

Selección del estudio

Las actividades de YoungProgramming fueron orientadas específicamente al trabajo con jóvenes los cuales son indicados para poner en práctica y evaluar la propuesta. Para llevar a cabo lo anterior, trabajamos con el grado décimo de la Institución Técnica Industrial sede Principal, la unidad de análisis para este caso son los equipos de trabajo conformados por 3 jóvenes.

El estudio de caso y los sujetos de investigación

Este estudio de caso es de tipo holístico [67] donde es considerada como unidad de análisis, jóvenes con edades comprendidas entre los 14 y 19 años, de grado décimo, quienes para efectos de este caso se constituyen como las fuentes primarias de información, en total son 18 jóvenes los cuales cursan el grado décimo.

Teniendo en cuenta las dinámicas que los jóvenes aplican en sus cursos académicos en la institución, se diseñó el estudio de caso con tres sesiones de trabajo en el grado Décimo que fueron desarrolladas desde el 05 de Septiembre de 2016 hasta el 07 de Septiembre de 2016, cabe resaltar que se varió un poco el orden de las actividades con respecto al orden propuesto inicialmente en el capítulo 3 de esta investigación (véase **Tabla 6**), esto con el fin de mejorar los tiempos de las actividades y agilizar el desarrollo de éstas. Las actividades quedaron de la siguiente manera:

Descripción del Caso

Actividad	Descripción	Sesión
Describir brevemente la actividad de aprendizaje	Se brindará a los jóvenes una pequeña introducción acerca del curso que se les va a impartir, se introducirá la herramienta Scratch con fines de exploración y realización de preguntas, además se explicarán los objetivos del presente trabajo, las prácticas de YoungProgramming y una muestra de lo que podrán alcanzar con el curso mediante el Mbot. Esto con el fin de entusiasmar y motivar a los jóvenes con las nuevas clases que van a recibir.	Primera Sesión
Exponer las unidades temáticas	Se impartirán las nociones básicas de programación, donde aprenderán los conceptos básicos de programación en Scratch.	Primera Sesión
Formar grupos de 3 estudiantes.	Se forman los grupos de tres estudiantes de forma aleatoria con ayuda del docente, garantizando que sean grupos homogéneos en cuanto a niveles de conocimiento.	Primera Sesión
Identificar al grupo con un nombre, logotipo, slogan	Una vez formados los grupos, los estudiantes deben identificarse mediante un nombre y un slogan, en esta actividad el nombre y slogan más creativo será premiado con puntos antes de iniciar la competencia.	Primera Sesión
Asignar roles	Se asignarán roles dentro de los grupos de la siguiente manera: un presidente, un vicepresidente y un secretario por cada grupo.	Primera Sesión
	Esta actividad cambia de orden con respecto al primer estudio de caso, ya que a los estudiantes se les	

Actividad	Descripción	Sesión
Especificar reglas de la actividad	<p>olvidaban las reglas justo en el momento de empezar la competencia ya que estas fueron propuestas con mucho tiempo de antelación.</p> <p>En esta actividad para la unidad de análisis se utilizaran las mecánicas de juego de puntos, desafíos, tablas de clasificación, beneficios y tiempos regresivos. Se deben presentar las reglas y restricciones dentro de la actividad para tener un control sobre comportamientos y circunstancias de la actividad.</p>	Segunda Sesión
Definir especificación de recompensas comunes para el grupo	<p>Esta actividad cambia de orden con respecto al primer estudio de caso, ya que a los estudiantes se les olvidaban las reglas justo en el momento de empezar la competencia ya que estas fueron propuestas con mucho tiempo de antelación.</p> <p>Se definirán recompensas que serán otorgadas siempre y cuando el grupo coopere para alcanzar los objetivos comunes de la clase. Para este estudio de caso, se entregarán recompensas a los dos primeros equipos que logren cumplir con la misión propuesta.</p>	Segunda Sesión
Formar grupos expertos(JigSaw)	<p>En esta actividad, cada uno de los integrantes de los grupos originales pasará a ser un nuevo integrante de un nuevo grupo “experto” (grupo de presis, vices o secres) con el objeto de que los alumnos desarrollen sus conocimientos sobre un tema determinado (solucionando un problema véase Anexos F, G, H) y que sean capaces de formular métodos eficaces de transmitírselos a otros.</p>	Segunda Sesión
	Los grupos de “expertos” (presis, vices y secres) se deshacen y los	Tercera Sesión

Actividad	Descripción	Sesión
Retornar a sus grupos originales	estudiantes pasan de nuevo a sus grupos originales “rompecabezas”, formado cada uno de ellos por alumnos que han llegado a dominar distintos subtemas.	
Programar en voz alta (RPPPVA & RoundRobin)	Cada integrante del grupo original programa (en voz alta), de acuerdo al conocimiento adquirido como “experto”, durante 5 minutos, compartiendo así su conocimiento con los demás (solucionando un problema véase Anexo E). Rotando hasta que todos los integrantes del grupo programen (RoundRobin). Este proceso de rotación se acaba cuando se logre solucionar el problema propuesto.	Tercera Sesión
Mantener el momento de colaboración	En esta actividad el docente y los observadores supervisarán cada una de las actividades propuestas y en caso de problemas intervendrán para que los grupos trabajen de forma adecuada.	Todas las sesiones
Realizar una evaluación sumativa	El objetivo de esta actividad es testear el nivel final de conocimientos (véase Anexo I) y verificar el desempeño de ciertos aspectos a nivel grupal (véase Anexo J).	Tercera Sesión
Hacer que los grupos comparen entre sí sus resultados	Esta actividad ha sido eliminada con respecto al anterior estudio de caso, ya que los estudiantes no lo veían como una retroalimentación sino como una oportunidad para criticar las soluciones de los demás.	-

Tabla 28. Actividades Estudio de Caso 2 – Guía LudiCoP

Diseño de los Indicadores y mediciones. Tomado de [69]

Para obtener la información necesaria para este Estudio de Caso, particularmente, y dar respuesta a la pregunta de investigación se tomaron un conjunto de mediciones e indicadores de [69]. La tabla 29 muestra un resumen de los indicadores y métricas definidos:

Pregunta	Indicadores	Mediciones	Instrumentos
¿Qué resultados se presentan en el comportamiento, la productividad y la calidad, en equipos trabajando con YoungProgramming en su versión refinada?	Resultado del nivel de productividad.	Nivel de comportamiento observado en los equipos de trabajo.	Observación, Protocolo de Observación (Anexo A), Misión (Anexo E)
	Resultado del nivel de comportamiento.		
	Resultado del nivel de calidad.	Nivel de Productividad observado en los equipos de trabajo.	Observación, Protocolo de Observación (Anexo B), Misión (Anexo E)
		Nivel de Calidad observado en los equipos de trabajo	Observación, Protocolo de Observación(Anexo C), Reporte de Errores (Anexo D), Misión (Anexo E)

Tabla 29. Tabla de Indicadores y mediciones.

Indicadores

Resultado del nivel de Productividad: Es el resultado de la productividad del grupo experimental (Décimo D) que realiza actividades y prácticas de YoungProgramming en su versión refinada. Cabe resaltar que los niveles de Productividad se encuentran en el rango de 0 a 100 por ciento.

Resultado del nivel de comportamiento: Es el resultado del comportamiento del grupo experimental (Décimo D) que realiza actividades y prácticas de YoungProgramming en su versión refinada. Cabe resaltar que los niveles de Comportamiento se encuentran en el rango de 0 a 100 por ciento.

Resultado del nivel de Calidad: Es el resultado de la calidad del grupo experimental la diferencia entre la calidad del grupo experimental (Décimo D) que realiza actividades y prácticas de YoungProgramming en su versión refinada. Cabe resaltar que los niveles de Calidad se encuentran en el rango de 0 a 100 por ciento.

Mediciones

Las mediciones de Comportamiento, Productividad y Calidad son iguales a las descritas en el diseño del estudio de caso número uno (EC1).

Resultados Cuantitativos

En este estudio de caso se conformaron los grupos de forma equitativa a consideración de los investigadores, de la siguiente manera:

Grupo Experimental: 6 grupos, cada uno con 3 integrantes conformados por una (1) mujer y dos (2) hombres.

A continuación, se presentan los principales resultados cualitativos del estudio de caso, en términos de productividad, comportamiento y calidad tanto para los equipos experimentales como para los equipos de control.

Productividad Grupo Final

Nombre del Equipo	Curso	Numero de respuestas de valor 5.0	Numero de respuestas de valor 4.0	Numero de respuestas de valor 3.0	Numero de respuestas de valor 2.0	Numero de respuestas de valor 1.0	Resultado Productividad (Fórmula 2)
Mario y sus Amigos	10d	7	2	0	0	1	88%
Panda	10d	9	0	0	0	1	92%
Seven Boom	10d	0	6	3	0	1	68%
Team Warriors	10d	9	0	0	0	1	92%
The Optical Boys	10d	8	1	0	1	0	92%
The Paints	10d	0	0	0	0	10	20%

Tabla 30. Productividad grupo final EC2

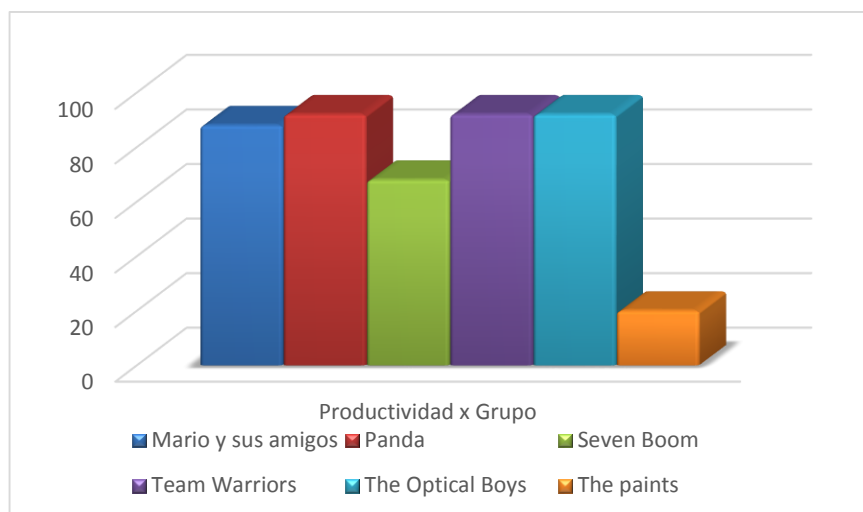


Figura 33. Productividad grupo final EC2

Calidad Grupo Final

Nombre del Equipo	Curso	Numero de respuestas Positivas (Si)	Numero de respuestas Negativas (No)	Resultado Calidad (Fórmula 3)
Mario y sus Amigos	10d	6	0	100%
Panda	10d	5	1	83%
Seven Boom	10d	4	2	66%
Team Warriors	10d	5	1	83%
The Optical Boys	10d	6	0	100%
The Paints	10d	0	6	0%

Tabla 31. Calidad grupo final EC2

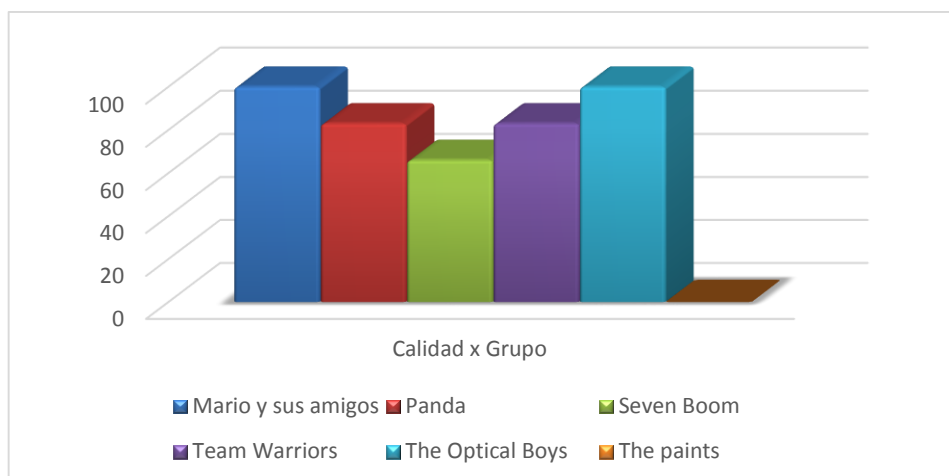


Figura 34. Calidad grupo final EC2

Comportamiento Grupo Final

Nombre del Equipo	Curso	Numero de respuestas satisfactorias	Numero de respuestas aceptables	Numero de respuestas insatisfactorias	Resultado Comportamiento (Fórmula 1)
Mario y sus Amigos	10d	5	2	0	85%
Panda	10d	7	0	0	100%
Seven Boom	10d	4	3	0	78%
Team Warriors	10d	6	1	0	92%
The Optical Boys	10d	5	2	0	85%
The Paints	10d	0	0	7	10%

Tabla 32. Comportamiento grupo final EC2

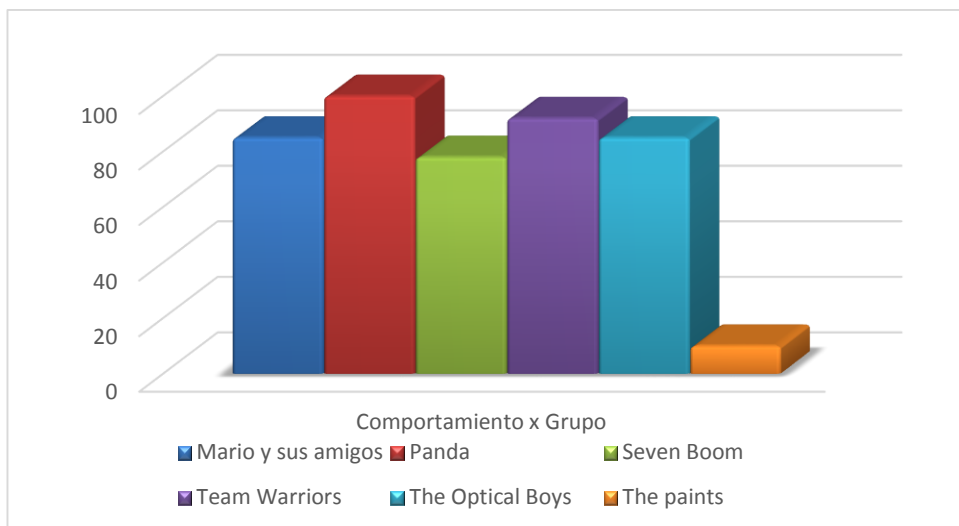


Figura 35. Comportamiento grupo final EC2

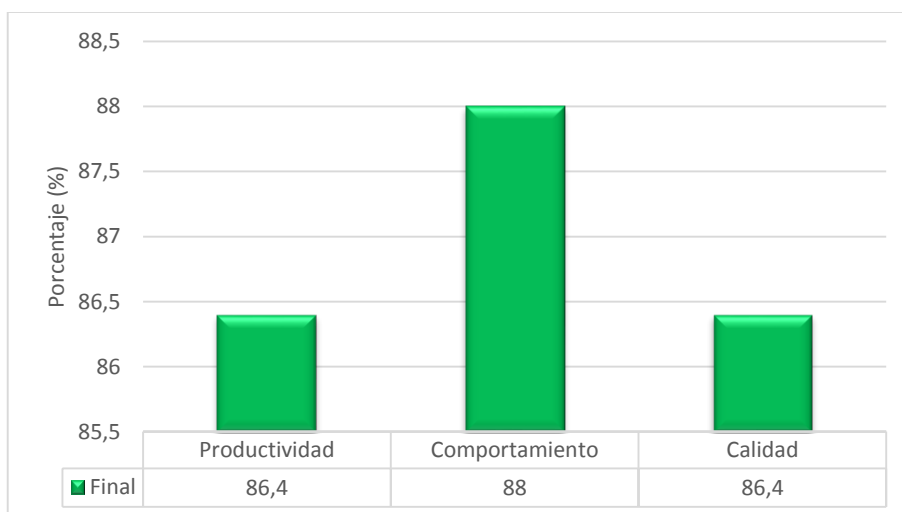


Figura 36. Porcentaje del Curso Final EC2

En la figura 36 se muestra el resultado del equipo final respecto a productividad, comportamiento y calidad.

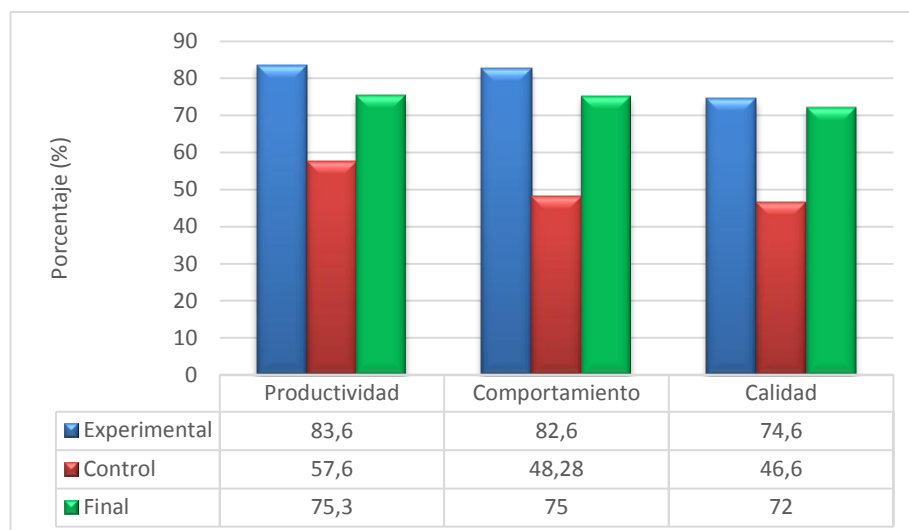


Figura 37. Mediciones del Curso Final EC2

En la **Figura 37** se puede encontrar una comparación realizada entre el primer estudio de caso (grupo de control y grupo experimental) y el segundo estudio de caso (grupo experimental final).

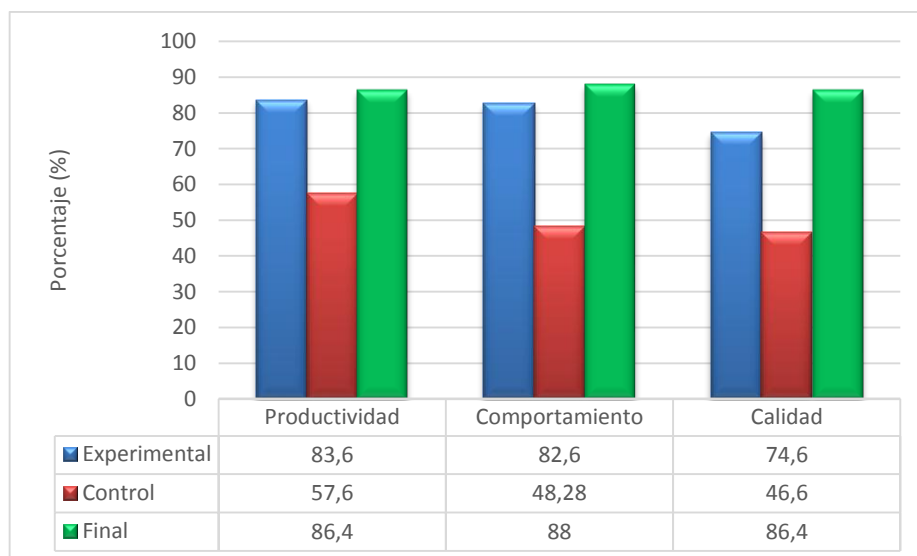


Figura 38. Mediciones del Curso Final sin grupo crítico EC2

En la **Figura 38** se muestra una comparación en donde se discriminan los datos arrojados por el grupo (Paints) ya que este no arrojó los resultados esperados y se consideró conveniente omitirlos en el análisis de este estudio. Con ello podemos observar que las mediciones durante el estudio de caso uno y dos fueron de manera incremental y positiva para este trabajo de investigación.

Evaluación de Habilidades Adquiridas

En este estudio de caso se realizó un test de habilidades (véase **Anexo I**) al grupo Experimental Final, con el fin de evaluar el aprendizaje después del uso de la estrategia. Este test, con preguntas tomadas de [71], arrojó como resultado lo siguiente:

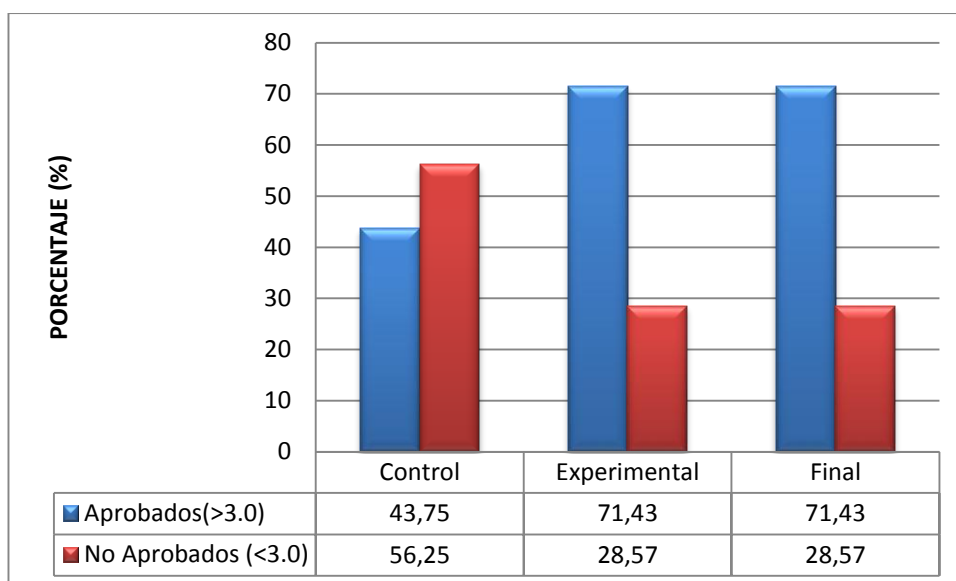


Figura 39. Porcentajes de Evaluación de Aprendizaje del curso (Experimental Final)

Una de las observaciones que cabe resaltar es el resultado del promedio de las evaluaciones entre los estudios de caso donde en el grupo de control uno se obtuvo un promedio de 2,9, posteriormente en el grupo experimental del estudio de caso uno se obtuvo un promedio de 3,54 y en el grupo experimental final del estudio de caso dos se obtuvo un promedio de 3,46; con lo cual se puede percibir que la estrategia YoungProgramming refinada aportó completitud al proceso de aprendizaje de la programación en jóvenes.

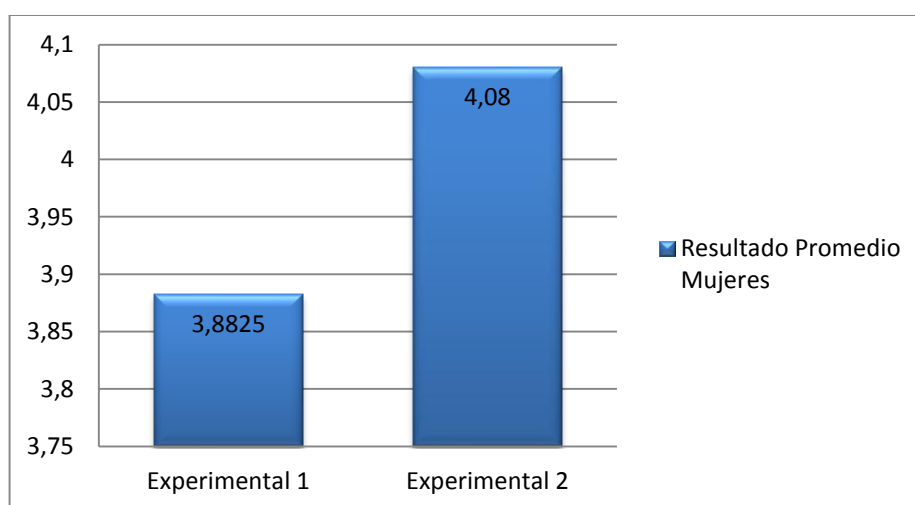


Figura 40. Porcentajes de Evaluación del Aprendizaje de Mujeres (Experimental EC1 y EC2)

Como observación adicional, al igual que en el estudio de caso anterior, se hace un análisis de las evaluaciones del aprendizaje de las mujeres dentro de los grupos que participaron en la actividad; se notó un incremento considerable que se puede observar en la **Figura 40**,

arrojando como resultado que las mujeres del grupo experimental final, al igual que en el primer estudio de caso con el grupo experimental, que utilizaron la estrategia YoungProgramming tuvieron un mejor desempeño y resultado en la realización de los Test.

Análisis de Encuestas

Al finalizar el estudio de caso se realizó, al igual que en el estudio de caso previo, una serie de preguntas tipo encuesta (véase **Anexo J**) las cuales fueron respondidas con indicadores de 1 a 5, con el fin de evaluar si los estudiantes percibían la existencia y presencia de las características y artefactos (Responsabilidad individual, Comunicación, Interdependencia Positiva, Igual Participación, Trabajo Colaborativo, Lúdica y Usabilidad) dentro de la estrategia YoungProgramming refinada, como resultado se pudo observar (véase **Figura 41**), que aunque no existió la presencia de estas características en su totalidad (igual a 5), si se percibió entre un 3,7 y un 4,4 la existencia de ellas. En cuanto a usabilidad los alumnos consideraron en un 3,7 que les pareció usable y que sería de agrado para ellos que esta estrategia fuera usada por sus docentes en caso de que ellos les enseñen a programar, al igual que en una de las características de la colaboración, los alumnos consideraron que la igual participación se vio reflejada con un promedio de 3,7.

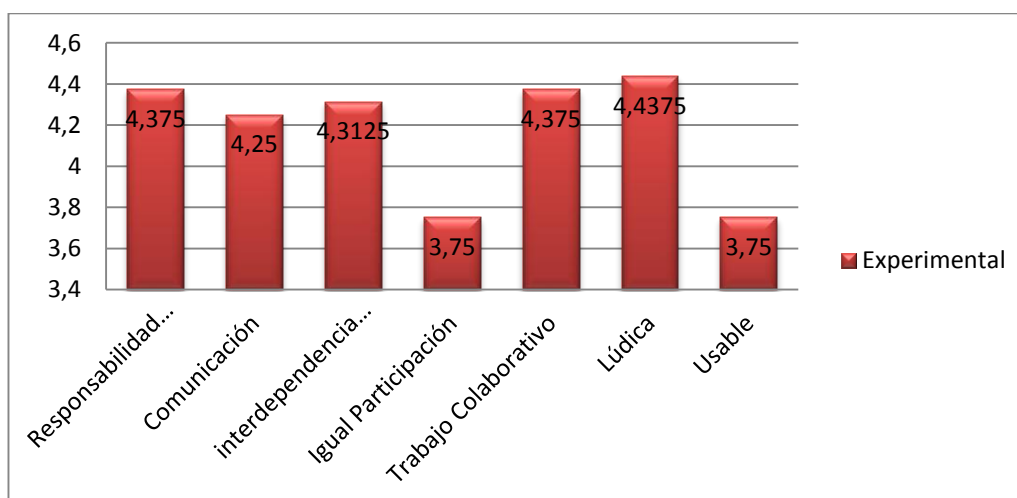


Figura 41. Resultados Encuestas (Experimental Final).

Análisis de Resultados

Análisis de Resultados Cualitativos

La **Figura 38** (Mediciones del curso final sin grupo crítico EC2) muestra una mayoría en los datos del curso experimental en las mediciones (productividad, comportamiento y calidad en la versión refinada de YoungProgramming) lo cual permite observar la influencia de la incorporación de la estrategia dentro del salón de clase que trata de aumentar el comportamiento, aprendizaje y el compromiso de manera lúdica sin que los jóvenes sientan la obligación de trabajar.

En este caso de estudio se vio una diferencia significativa en la actitud y el comportamiento de algunos jóvenes quienes desde un comienzo rechazaron totalmente la actividad y a diferencia de sus compañeros no entregaron resultados. Por lo tanto, es necesario generar una motivación adicional (notas para la materia) a los grupos críticos con el fin de que participen de manera indicada en la actividad. Además, es necesario en todo proceso colaborativo realizar un trabajo previo, es decir, ambientar a las personas en actividades colaborativas, con finalidad de que se familiaricen con éstas.

Capítulo 6

6.1 CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

A continuación, se describen las conclusiones del presente trabajo de investigación, sus limitaciones y trabajos propuestos a futuro.

6.2 RESUMEN DEL PROYECTO

Este trabajo es el resultado de probar técnicas colaborativas y lúdicas aplicadas bajo un método de análisis y aplicación de la colaboración y actividades lúdicas en un entorno educativo, en el cual se permite a jóvenes bachilleres de últimos años, asimilar conceptos de programación a través de la incorporación de prácticas lúdicas, ágiles, cognitivas y colaborativas adaptadas en un entorno de aprendizaje conformado por jóvenes.

Para ello, se adaptó el modelo ChildProgramming mediante la reformulación de algunas de sus prácticas, roles y artefactos donde se consideraron además, conceptos lúdicos, mecánicas de juego, dinámicas de juego, propios del área de la Lúdica, de tal manera que los resultados en compromiso, calidad, comportamiento y motivación en los grupos de jóvenes influyan directamente en el desempeño de las actividades de programación. Como resultado de este trabajo se caracterizó la estrategia YoungProgramming y se creó una técnica de aprendizaje colaborativo llamada LudiCoP, la cual hace parte del proceso YoungProgramming.

En el estudio de caso número uno se obtuvieron resultados de comportamiento, calidad y productividad de dos grupos de estudio, uno trabajando bajo el enfoque de aprendizaje tradicional y el otro apoyado por la estrategia YoungProgramming.

Para el caso de estudio número dos se obtuvieron resultados también de comportamiento, calidad y productividad, de un grupo trabajando bajo una versión de la estrategia YoungProgramming refinada, la cual ha sido ajustada a partir del resultado obtenido en el estudio de caso número uno.

6.3 CONCLUSIONES

La principal conclusión de este trabajo es que la incorporación de aspectos lúdicos y colaborativos incrementan el desempeño de los equipos de trabajo. En los casos desarrollados se muestra como la lúdica incentiva el aprendizaje de los jóvenes, los cuales se ven identificados con las mecánicas de juego y ven el aprendizaje de manera divertida y competitiva.

Al trabajar por grupos, los jóvenes se apropian y socializan el conocimiento de la programación tal como se vio en las actividades colaborativas, apropiándose de definiciones como, por ejemplo: programa informático, algoritmo, lenguaje de programación, variable, estructura básica de programación, estructura de control y ciclo. Los jóvenes participan activamente con sus compañeros por un beneficio común. Ellos ven en el aula de clase un juego en el que van ganando puntos y otros reconocimientos. Se mezcla el aprender con el jugar al agregar los elementos de juego pertenecientes a la lúdica en el entorno de aprendizaje o al mezclar dichos elementos con las actividades pertenecientes a su aprendizaje.

En la estrategia YoungProgramming no solo el Tutor se encarga de transmitir conocimientos, los alumnos también son capaces de crear mecanismos para transmitir su conocimiento a los demás integrantes del equipo ya que éstos se vuelven “expertos” en una determinada temática gracias a la incorporación de técnicas de aprendizaje colaborativo.

En los estudios de caso desarrollados se evidencia la gran diferencia en términos de productividad, calidad y comportamiento de los equipos que trabajaron bajo el enfoque YoungProgramming y los que lo hicieron bajo el enfoque de aprendizaje tradicional. Además, se logra observar que los equipos que trabajaron con la estrategia YoungProgramming

obtuvieron mejores resultados en el test, es decir, que a nivel individual el nivel de conocimiento adquirido es mayor en los equipos que trabajaron con YoungProgramming.

Uno de los resultados que se debe resaltar es el mejoramiento en el desempeño (test de habilidades) de participación de las mujeres que fueron parte de las actividades desarrolladas con la estrategia YoungProgramming lo cual da un indicio de que esta aportó significativamente en los resultados obtenidos.

El aprendizaje y el trabajo colaborativo requieren de un prolongado proceso, por lo tanto no se puede pretender alcanzar resultados excepcionales es unas pocas sesiones de trabajo, si no que al contrario, este es un procedimiento que requiere evolución a través del tiempo.

6.4 RECOMENDACIONES

Para trabajos posteriores que se realicen basados en este proyecto, se desea en lo posible que se tenga presente durante el proceso de investigación las siguientes recomendaciones:

Debido a que en este trabajo de investigación se planteó tomar como población de estudio jóvenes de bachillerato entre los 14 y los 19 años de edad se recomienda planear las actividades de trabajo grupal de manera clara y sencilla, utilizando un lenguaje adecuado que les facilite la comprensión de los conceptos tratados en el proceso o en las actividades, ya que el tiempo del que se dispone en los colegios suele ser limitado y para evitar posibles inconvenientes de entendimiento que retrasen el correcto desarrollo de las actividades.

Una de las complicaciones que se tuvieron en los dos estudios de caso, es que los alumnos ya llegaban cansados a la realización de encuestas y test de habilidades. Se recomienda que la evaluación no sea realizada de manera inmediata, sino posteriormente, puede ser un día después de las actividades.

Otra de las complicaciones que se presentó es el déficit de atención por parte de los estudiantes que trabajaron en las jornadas de la mañana. Por lo tanto es recomendable establecer jornadas de trabajo en otras horas, preferiblemente en las cuales no deban madrugar.

Se recomienda la realización de un pre-test de conocimientos con el fin de establecer el nivel de conocimiento en el área de la población, para realizar un mejor proceso de selección de equipos de trabajo.

Con el fin de agilizar los tiempos de las actividades es necesario redefinir la **Tabla 6** con el orden establecido en el Estudio de Caso número dos (véase **Tabla 28**).

En todo proceso colaborativo hay que realizar un trabajo previo, es decir, ambientar a las personas en actividades colaborativas, con finalidad de que se familiaricen con éstas y puedan realizar un trabajo de manera más fluida.

6.5 TRABAJO FUTURO

- Explorar el desarrollo de estudios de caso en otros niveles de educación, por ejemplo en el nivel universitario con misiones de mayor dificultad que utilice la estrategia YoungProgramming para que influya directamente en la motivación de las personas.

- Explorar el desarrollo de estudios de caso donde la unidad de análisis sean mujeres, con el fin de contrastar la influencia de género en el aprendizaje de la programación bajo el enfoque YoungProgramming.
- Establecer el uso de la estrategia para el desarrollo del pensamiento computacional en las escuelas para toda la secundaria y así esta propuesta no quede tan solo en una estrategia de aprendizaje, y pueda estar inmersa en un pensum educativo.
- Crear una herramienta que permita llevar un flujo de trabajo de las actividades realizadas bajo el enfoque YoungProgramming.

7. REFERENCIAS BIBLIOGRAFICAS

- [1] Hurtado, J., Collazos, C., Cruz, S., Rojas, O. (2011). ChildProgramming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Lúdica, la Colaboración y la Agilidad. Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca. Popayán, Colombia
- [2] Kumar, B. & Khurana, P. (2012). Gamification in education – Learn computer programming with fun.
- [3] Dillenbourg, P. (1999). What do you mean by collaborative learning, Collaborative learning: Cognitive and computational approaches.
- [4] Johnson, D. and R. Johnson. (1984). Circles of Learning, Washington, DC: Association for Supervision and Curriculum Development.
- [5] Collazos, C., Guerrero, L., Pino, J., and Ochoa, S. (2002). "Evaluating Collaborative Learning Processes". Proceedings of the 8th International Workshop on Groupware (CRIWG 2002), Springer Verlag LNCS, 2440, Heidelberg, Germany, September.
- [6] Roschelle, J., & Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. In C. O'Malley (Ed.), Computer-supported collaborative learning (pp. 69-197). Berlin, Germany: Springer Verlag.
- [7] Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences. Cambridge, UK: Cambridge University Press.
- [8] Norman, D.A. (1993). Things that make us smart: defining human attributes in the age of the machine. NY: Addison-Wesley.
- [9] Bravo-Lillo, C. (2015). Pensamiento computacional: una idea a la que le llegó el momento. Revista Bits de Ciencia N°12. Departamento de Ciencias de la Computación, Universidad de Chile, Santiago de Chile, Chile.
- [10] Kelleher, C., Pausch, R. (2005). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers, ACM Computing Surveys, 37 (2):83-137.
- [11] Wing, J. M. (2006) 'Computational Thinking', Communications of the ACM, vol. 49, no. 3, pp. 33–5.
- [12] Robinson, K. (2006). How schools kill creativity. TED Talk. Disponible en: http://www.ted.com/talks/ken_robinson_says_schools_kill_creativity. Consultado el 3/septiembre/2015.
- [13] Meerbaum-Salant, O., Armoni, M., Ben-Ari, M. (2010). Learning Computer Science Concepts with Scratch. Department of Science Teaching Weizmann Institute of Science.

- [14] Javidi, G., Sheybani, E. (2009). “Digispired: Digital inspiration for interactive game design and programming”, *Journal of Computing Sciences in Colleges*, vol. 24, núm.3, pp. 144-150.
- [15] Jurado, F., Molina, A., Redondo, M., Ortega, M. (2012). *Cole-Programming: Incorporando Soporte al Aprendizaje Colaborativo en Eclipse*. IEEE-RITA Vol. 7, Núm. 3.
- [16] Chen-Chung, L., Yuan-Bang, C., Chia-Wen, H. (2011). The effect of simulation games on the learning of computational problem solving. Graduate Institute of Network Learning Technology, National Central University, Taiwan.
- [17] Posada, R. (2014). *La lúdica como estrategia didáctica*. Universidad Nacional de Colombia, Colombia.
- [18] Runeson, P. & Höst, M. (2009) Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*.
- [19] M. Hicks, J.S. Foster. (2008) “Adapting Scrum to Managing a Research Group”.
- [20] Schwaber, K. & Beedle, M. (2001) *Agile Software Development with Scrum (1sted.)*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [21] Consorcio de Habilidades Indispensables para el Siglo XXI. (2009). *Logros indispensables para los estudiantes del siglo XXI*.
- [22] Fracchia, C., Alonso, A., Martins, A. (2014). Enseñanza de la programación: un tema en la agenda académica para repensar año a año. XX Congreso Argentino de Ciencias de la Computación. Buenos Aires, Argentina.
- [23] Wolz, U., Leitner, H., Malan, D., Maloney, J. (2009). “Starting with scratch in cs1”, en *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, Chattanooga.
- [24] Rizvi, M., Humphries, T., Major, D., Jones, M., Lauzun, H. (2011). “A cs0 course using Scratch”, *Journal of Computing Sciences in Colleges*, vol. 26, núm. 3, pp. 19-27.
- [25] Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller, P. (1998). *Mini-languages: a way to learn programming principles*, in *Education and Information Technologies* Vol. 2, pp. 65 -83.
- [26] Herramienta Scratch. [Online]. Available: <http://scratch.mit.edu/>
- [27] Piaget, J. (1978). *La representación del mundo en el niño*. Madrid: Morata.
- [28] Vygotsky, L. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.
- [29] Silva, C., Campos, R., Ramo, O. (2003). *Método María Montessori, Psicología Educativa* Valdebenito, Instituto Profesional Luis Galdanes.
- [30] King, A. (1991). "Effects of training in strategic questioning on children's problem-solving performance". *Journal of Educational Psychology*.

- [31] Kolfschoten, G., Briggs, R., Vreede, G. (2006). "Definitions in Collaboration Engineering".
- [32] Bruffee, A., Kenneth, A. (1999). Collaborative Learning: Higher Education, Interdependence, and the Authority of Knowledge.
- [33] Smith, K. (1996). "Cooperative learning: making "group work", " Using active learning in college classes: A range of options for faculty, *New directions for teaching and learning*, vol. 67.
- [34] Johnson, D., Johnson, R., Holubec, E. (1998). Cooperation in the classroom.
- [35] Brown, J. (1983). "Process versus product: a perspective on tools for communal and informal electronic learning". *Report from the Learning: Education in the Electronic*.
- [36] Boder, A. (1992) "The process of knowledge reification in human-human interaction" *Journal of Computer Assisted Learning*, Vol. 8, No. 3, September, pp. 177-185.
- [37] Johnson, D., Johnson, R. (1987). Learning together and alone.
- [38] Collazos, C., Mendoza, J. (2006). "Cómo aprovechar el aprendizaje colaborativo en el aula". *Revista educación y educadores*, pp. 61-76.
- [39] Cooper, J., Prescott, S., Cook, L., Smith, L., Mueck, R., Cuseo, J. (1990). "Cooperative learning and college instruction: Effective use of student learning teams".
- [40] Sapon, M. (1998). "Because we can change the world".
- [41] Dreves, C. (1984). "¿Qué es el aprendizaje cooperativo? Extractado de: Circle of Learning, Johnson, D. & Johnson, R.".
- [42] Lucero, M. (2003). "La colaboración y el aprendizaje colaborativo en los ambientes virtuales de aprendizaje". Universidad Nacional de San Luis Argentina.
- [43] Pantoja, W., Collazos, C., Penichet, V. (2012). CSCW aplicado a la mejora de procesos de software.
- [44] Barros, B. & Verdejo, M. (2001). "Entornos para la realización de actividades de aprendizaje colaborativo a distancia". Departamento de lenguas y Sistemas Informáticos.
- [45] Roschelle, J & Teasley, S. (1995). "The construction of shared knowledge in collaborative problem solving". *Computer supported collaborative learning*, pp. 169-197.
- [46] Soller, A. (2001). "Supporting social interaction in an intelligent collaborative learning system".
- [47] Johnson, D & Johnson, R. (1978). "Cooperative, competitive, and individualistic learning". *Journal of Research and Development in Education*, vol. 12, pp. 8 –15.
- [48] Dillenbourg, P., Baker, M., Blake, A., Malley, C. (1995). "The evolution of research on collaborative learning".

- [49] Gonzáles, M. (2003). “Modelo de enseñanza-aprendizaje en situaciones no presenciales: un estudio de caso”. Tesis doctoral. Universidad de Murcia.
- [50] Gómez, S. (2012). Modelo para la selección de técnicas de aprendizaje colaborativo soportado por computador. Universidad Nacional de Colombia. Medellín, Colombia.
- [51] Ramírez, D., Bolaños, J. (2013). Guía para el diseño de actividades de aprendizaje colaborativo asistido por computador. Universidad del Cauca. Popayán, Colombia.
- [52] Kagan, S. (1994) Cooperative Learning. San Clemente, CA: Kagan Publishing.
- [53] Barkley, E., Cross K.P., Major, C. Técnicas de aprendizaje colaborativo: manual para el profesorado universitario. Ediciones Morata: Madrid, 2007.
- [54] Molina, A. (2007). CIAN y CIAM. Un marco conceptual y metodológico para el diseño de aplicaciones colaborativas e interactivas. Journal of Universal Computer Science 14 (9), 1435-1446.
- [55] Gea, M., Gutierrez, F., Garrido, J., Cañas, J. (2003). AMENITIES: Metodología de Modelado de Sistemas Cooperativos. Universidad de Granada, España.
- [56] Gardner, M (1979). Circo matemático. Alianza Editorial. Madrid.
- [57] Herranz, E. (2013). Gamification, I Feria Informática .Universidad Carlos III Madrid España.
- [58] Butler T. (1988). Games and Simulations: Creative Educational Alternatives. Volumen 33, pp. 20-28.
- [59] Gairín J.M (1990). Efectos de la utilización de los juegos educativos. Centro de profesores. Zaragoza.
- [60] Alhadeff, E. (2007). Serious Games – A sizeable market. Blog Future-making Serious Games.
- [61] Guerrero, D., Gomez, H., Anaya, R. (2009). Juegos en la Enseñanza de la Ingeniería de Software. Revista Tecnológicas No.22.
- [62] Beza, O. (2011). Gamification – How games can level up our everyday life?. Universidad de Amsterdam, Holanda.
- [63] Cortizo, J., Carrero F, Pérez J. (2011). Gamificación y Docencia: Lo que la Universidad tiene que aprender de los Videojuegos. En VIII Jornadas Internacionales de Innovación Universitaria 2011, Universidad Europea de Madrid.

- [64] Werbach, K., & Hunter, D. (2012). For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press.
- [65] Huizinga, Johan (1938) Homo Ludens. Beacon Press
- [66] Runeson, P. & Höst, M. (2009). "Guidelines for conducting and reporting case study research in software engineering". Empirical Software. Engineering.
- [67] Yin, R.K. (1984). Case Study Research. Design and Methods. Beverly Hills, Sage.
- [68] Chetty, S. (1996). The Case Study Method for Research in Small and Medium sized firms.
- [69] García, A. & Orejuela, H. (2014). ChildProgramming-G: Extendiendo ChildProgramming con Técnicas de Gamificación. Universidad del Cauca.
- [70] Ministerio de Educación Nacional de Colombia. [Online]. Available: <http://www.mineducacion.gov.co/1759/w3-article-244739.html>
- [71] López, Juan. (2010). Programación con Scratch. Fundación Gabriel Piedrahita Uribe. Tercera Edición.
- [72] Preocupante déficit de ingenieros en Colombia. Periódico El Tiempo. [Online]. Disponible en : <http://www.eltiempo.com/estilo-de-vida/educacion/panorama-de-los-ingenieros-en-colombia/16402298>